Università degli Studi di Padova

FACOLTÀ DI INGEGNERIA
Corso di Laurea Magistrale in Ingegneria Elettronica

Tesi Magistrale

Analysis and development of a Motor Field Oriented Control algorithm for 3ph PMS Brushless Motors using Cortex-M4 microcontroller

Candidato:
Giovanni Trojan
Matricola 1148496

Relatore:
Matteo Meneghini

Correlatore:
Matteo Vitturi

Anno Accademico 2018–2019

Abstract - EN

In a world that increasingly looks at electric power sources and energy savings, this thesis has the aim of developing an efficient control for three-phase electric motors, the Field Oriented Control, using the Infineon ARM M4 microcontroller family. The theoretical scheme has been analyzed and adapted for a more performing software implementation and, subsequently, performance tests have been done, confirming the precision, speed and efficiency of the control and positioning it in the modern perspective of saving energy resources.
The software has been developed with a modular architecture with the aim of being able to easily make changes or new improvements, while portability guarantees the compatibility of the program with any board equipped with Infineon ARM-M4 microcontroller.

Abstract - IT

In un mondo che guarda sempre più all'elettrico e al risparmio energetico, questa tesi ha l'obiettivo di sviluppare un efficiente controllo per motori elettrici trifase, il Field Oriented Control, per la linea di microcontrollori Infineon ARM M4.
Lo schema teorico è stato analizzato ed adattato per una più performante implementazione software e, successivamente, sono stati effettuati test sulle prestazioni, confermando la precisione, la velocità e l'efficienza del controllo e posizionandolo nella moderna ottica del risparmio delle risorse energetiche.
Il software è stato sviluppato con un'architettura modulare con l'obiettivo di poter aggiungere con semplicità, modifiche o nuove migliorie, mentre la portabilità garantisce la compatibilità del programma con qualsiasi board con microcontrollore Infineon ARM-M4.

"I can think of nothing else than this machine."
James Watt

Contents

List of Figures

1 Introduction

This thesis was entirely realized during an internship at the Padua Development Center of the company Infineon Technologies, working as a member of the Microcontroller (MC) team. This chapter has the aim to introduce the company and its philosophy as well as the MC team itself. After that, the problem statement and the thesis goals are presented.

1.1 Infineon Technologies

Infineon Technologies is a leading innovator in the international semiconductor industry founded in April 1999. It is head quartered in Munich, Germany, but it has 17 Production sites and 35 Research and Development sites scattered all over Europe,the Americas and the Pacific Regions, and more than 40000 employees worldwide.

Figure 1.1: Infineon's Logo

Today the company is working on 4 business areas. Here follows the official presentation given by the company for these 4 areas:

- Automotive (ATV): In the Automotive (ATV) segment, we develop products and solutions for conventional drivetrains while also actively shaping the keystone trends that define the industry. Demand for our power semiconductors is on an upward path, fueled by the rising number of electronic applications in cars – a trend further accentuated by the growing popularity of electromobility. We are the undisputed market leader in silicon-based IGBTs and IGBT modules. Our expertise in silicon carbide is also increasingly relevant for automotive power semiconductors. We are paving the way for self-driving cars with our radar sensors and microcontrollers. Positioned as number two in the radar sensor market, we are already noting strong momentum from the proliferation of driver assistance systems. In the long term, radar systems will be fused with other sensor technologies. We are laying the groundwork for this by developing products such as LIDAR solutions. With our AURIX™ family, we are also benefiting from the trend towards increased automation. Our products here control electronic systems such as steering and braking, also acting as host controllers to provide functional safety and data security for central computing platforms.

- Industrial Power Control (IPC): The Industrial Power Control (IPC) segment specializes in the efficient conversion of electrical energy along the entire supply chain – from generation and transmission right through to consumption. Applications here include photovoltaic, wind turbines, high-voltage DC transmission systems, energy storage systems, charging infrastructures for electric vehicles, trains and home appliances. Infineon is the world leader in IGBT-based power semiconductors and modules. To further strengthen this core IPC business, we are aiming for technology leadership in silicon carbide (SiC). Complementary to this is our strong belief in integration and digitalization. Both the IPMs (Intelligent Power Modules) from our CIPOS™ family and the motion control solutions from our iMOTION™ family enable our customers to use energy more efficiently.

- Power Management and Multimarket (PMM): Our Power Management & Multimarket (PMM) segment focuses on power semiconductors for energy management as well as components for wireless infrastructures and mobile devices. PMM also specializes in ultra-reliable components for applications in industries such as aerospace. Infineon is the clear leader in the global MOSFET market. Our CoolMOS™ and OptiMOS™ families deliver excellent levels of energy efficiency. We also offer leading-edge solutions based on gallium nitride. In parallel to this product group, we are continuing to expand our portfolio of complementary drivers and controllers. Battery-operated devices are one of the fastest-growing applications for power semiconductors. In the high-frequency and sensor space, we have established a strong technology footprint with MEMS microphones (silicon in particular), time-of-flight sensors for 3D cameras and radar applications. We have already established very successful positions in the respective markets. At the same time, we can apply our expertise in these areas to more and more use cases that are set to gain momentum over the coming years. Key examples here include human-machine interaction (HMI) and facial recognition.

- Digital and Security Solutions (DSS): The Digital Security Solutions (DSS) segment has over thirty years' experience delivering some of the world's most challenging and large-scale digital security projects. Our success here is built on our wealth of expertise in conventional smart card applications. We are transferring our core skills in payment cards and government documents to the fast-growing field of embedded security applications. As digitalization shapes more and more areas of everyday life, security is becoming a key success factor for applications across industries as diverse as computing, automotive, Industry 4.0 and smart homes. Parallel to its role as an independent business segment, DSS acts as a competence center for our other three segments, supporting their efforts to hardwire security functionality into their respective system solutions.

Working on all these fields Infineon keeps tight relationships with many customers. Some of them are shown in Figure 1.2.

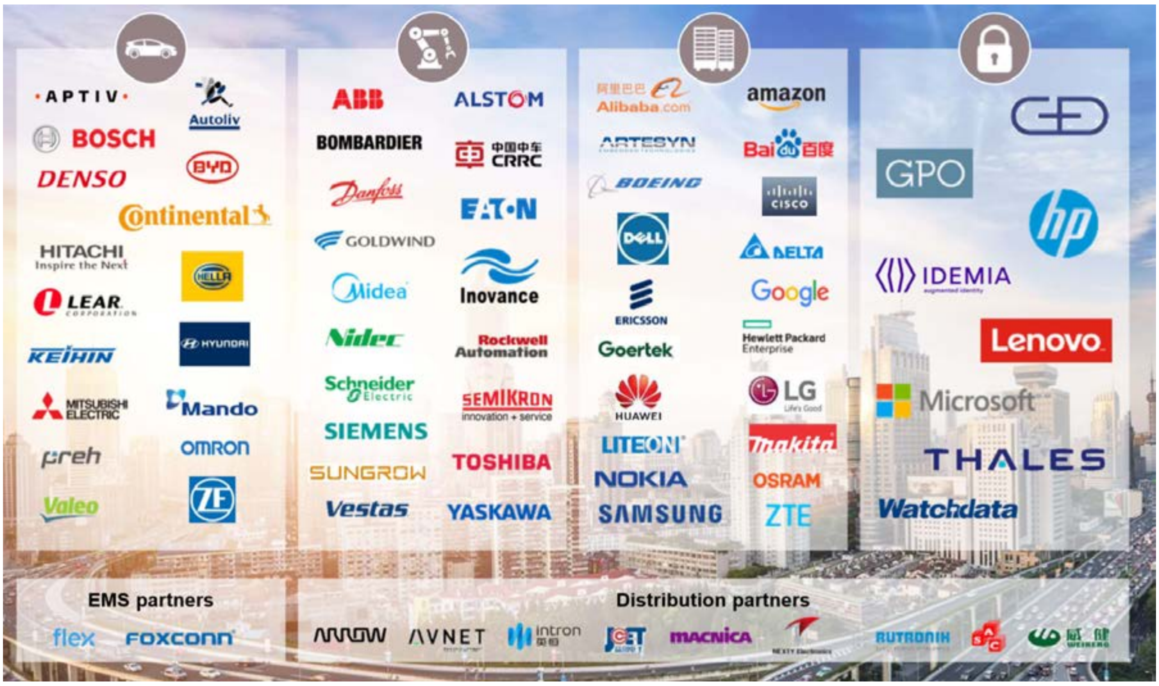


Figure 1.2: Infineon's Customers

1.1.1 Automotive MC Team in Padua

Located in the Development Center of Padua, the Microcontroller (MC) team is part of the product and testing engineering organization of ATV, focusing on embedded Flash of automotive and industrial microcontrollers.
The team has different responsibilities:

- It contributes to Non-Volatile Memory (NVM) testing concept, analyzability and manufacturability, validation and analysis planning;
- It provides NVM analysis tools to improve automation;
- It provides embedded firmware and test patterns for productive testing on NVM;
- It performs NVM test-chip analysis for design and technology learning;
- It executes NVM validation and characterization;
- It performs enhanced In-System tests on application conditions;
- It provides Design Validation Reports for Customer presentations;
- It supports NVM qualification (product and technology);
- It reviews test program contents of test package releases (production, qualification, characterization).
- It supports a fast design-in of Infineon Products into customers application
- It also prepares ready-to-use supplementary low-level software for the costumers to reduce their time to market

To better respond to these tasks, Padua MC team is divided into two sub-teams:

- *Test Engineering*: this group develops embedded software for analysis and testing. In particular, this group is responsible for development of the Firmware and of test program used for performing Flash memory tests.
- *Product Engineering*: this group characterizes embedded flash and it validates robustness of non volatile memories. Its purpose is ensuring the quality of memory by validating customers requirements, also considering statistical aspects, extending validation to many samples. It analyzed the problem, it searches its causes in order to find a solution and therefore it offers this solution to the designers and technology experts.
  The characterization activity tries to push parameters, like temperature, power and system frequencies, to the limits.

- *ACE*: the Ease-of-Use Project has been founded in order to enable additional business with Infineon products by addressing the needs of a broader market towards mass market and increase the efficiency of our support structures.
  The output of this project is addressing customers hardware developer, software developer, system architects and tool vendors with no or less know-how on Infineon products architecture and their eco-system.

- *SW*: this group is responsible for the development of integrative software for Infineon hardware, guaranteeing the client less development costs and a lower time-to-market.

1.2 Motor control history

Ever since the first steam engine appeared, the problem of being able to control rotation speed has arisen. One of the first and greatest motor speed regulator was introduced in 1788 by James Watt, that implemented into the Newcommen engine the famous centrifugal governor, a real feedback system that controls the speed of the engine by regulating the flow of working fluid, so as to maintain a near-constant speed.

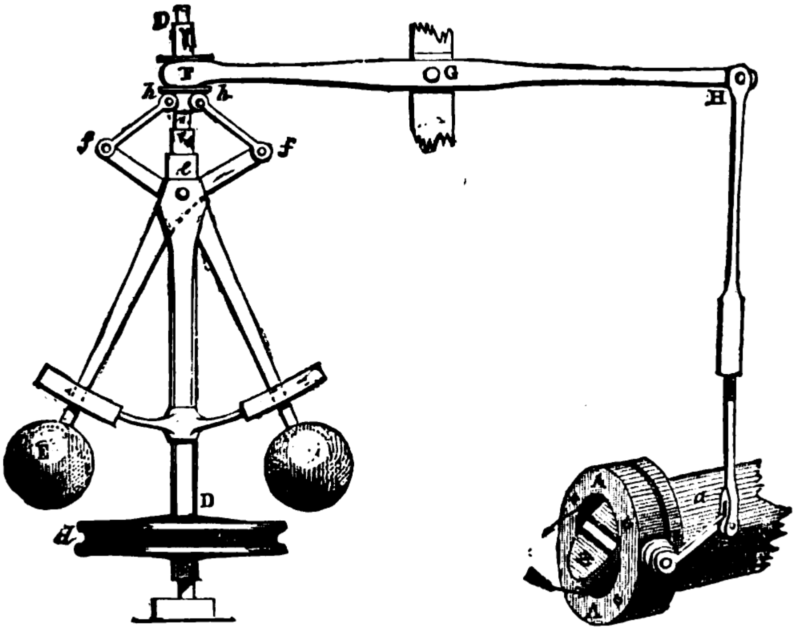


Figure 1.3: Centrifugal governor and throttle valve

At the beginning of the nineteenth century first electric motors began to appear, thanks to the work of many inventors. Consequently, the challenge of controlling motor speed continued in this kind of motors, with different schemes and methods.
In conjunction with advances in microelectronics technology and power semiconductor devices since the 1980s, rapid progress has been made in the field of power supply and drive control. Consequently, various control functions have been developed tailored to specific applications, also taking into account efficiency.
The evolution of the electric motor has also been underpinned by design, production, and material technologies and, in recent years, motor performance has received a major boost from a variety of simulation techniques and studies in materials technology, particularly in the field of permanent magnet motors. Prompted by concerns about reducing environment problems, in addition to the expected further improvements in efficiency, the future will also see further research into motor structure aimed at reducing material usage through smaller motor sizes and making available materials that do not include scarce resources[2]. With use of electric motors and power supply becoming standard practice, it is also anticipated in [2] that motors, power supply and motor control will become integrated in both a structural and design sense.
The electric motor has become a key device that underpins society in a wide range of fields including power generation, industry, transport, and home appliances. Moreover, generators and electric motors are widely used as power sources in factories and railways, and even in vehicles and data processing equipment. It is also widely known that motors account for approximately 40% of all electricity consumption[1]. As well as playing a major role in industrial progress, it is no exaggeration to say that rotating machinery in the form of motors and generators represent part of the foundations of modern society.

1.3 Objectives of this thesis

The aim of this thesis is to implement a digital control scheme called Field Oriented Control for Permanent Magnet Synchronous Motors using ARM M4 microcontroller. The position and speed are estimated starting from motor phase voltages and currents, therefore, position sensors or hall probes are not used ensuring greater cost savings.
The best hardware that meets the needs of the control scheme is selected and the software is developed to guarantee a fast and adaptable control to multiple boards.
Measures of control performances are then taken to verify the successful implementation.

2 Motors

An electric motor is a machine that converts electrical energy into mechanical energy. Motors are divided into two families: motors powered by direct voltage (DC) and motors powered by alternating voltage (AC, three-phase or single-phase).
Each motor is composed of these basic components:

- *Rotor*: In a electric motor, the moving part is the rotor and it is connected to a shaft to transfer mechanical energy. The rotor usually has conductors laid into it that carry currents, which interact with the magnetic field of the stator to generate the forces that turn the shaft. Also, there are some rotors that can carry permanent magnets, and the stator holds the conductors.

- *Bearings*: the rotor is held by bearings that guarantee freedom of rotation with very little friction. The motor shaft extends through the bearings to the outside of the motor, where the load is applied.

- *Stator*: the stator is the stationary part of the motor's electromagnetic circuit and can consist of coils or magnets, depending on the type of motor.

- *Coils*: the coils are cable windings usually wrapped around a laminated soft iron magnetic core so as to form magnetic poles when energized with current. They can be mounted on the stator or on the rotor.

The distance between the rotor and stator is called the air gap. The air gap has important effects, and it is generally as small as possible, as a large gap has a strong negative effect on performance, in fact it is the main source of the low power factor at which motors operate.
Also, each type of motor has its own peculiar components.
Electric generators are mechanically identical to electric motors, but operate in the reverse direction, converting mechanical energy into electrical energy. Taking this concept into account, the interactions between the coils and the relative rotating magnetic field of the magnets caused by rotor movement generate an additional voltage to the electrical ends of the motor called back electromotive force (BEMF).

2.1 DC Motor

DC motors generally have windings on the rotor and permanent magnets on the stator. They were the first electric machine made, and they are still widely used for small and large powers, as a generator or motor.
There are small power motors for domestic use, as well as engines for railway and marine traction with power of many hundreds of kW. As the name implies, this kind of motors is powered with a continuous voltage.
It is necessary to have an additional device that switch the current flow in the windings of the motor in order to generate a rotating magnetic field. To fulfill this requiement, the DC electric motors have an internal mechanism, either electromechanical or electronic, that periodically change the direction of current flow in the windings.
There are two types of approaches to implement this device:

- *Brushed Motors*: as the name implies, they are characterized by the presence of brushes, electrical connections in contact with the shaft. Depending on their position, they can make the circuit close differently, alternating the polarity in the coils every half turn. The advantages of this kinds of motors are the low cost and a simple wiring. In fact, the brushed motors can be connected directly to the DC power supply and the control can be as simple as a switch.
  On the other hand, these motors bring also disadvantages: lower efficiency, high electrical noise level since the switching action that constantly closes and opens the inductive circuits creates a large amount of electrical and electromagnetic noise and shorter life, because the brushes are in perennial physical contact with the shaft and are subjected to friction.

- *Brushless Motors*: Unlike brushed motors, they do not need to have sliding contacts (brushes) on the rotor shaft to work, the switching of the current circulating in the stator windings and therefore the variation in the orientation of the magnetic field generated by them occurs electronically. Since the controller must know the position of the rotor with respect to the stator in order to determine the orientation to be given to the magnetic field, it is usually connected to a Hall effect sensor or a hollow shaft resolver.
  Brushless motors have a longer life span and high efficiency because of the absence of brushes but they are usually more expensive.

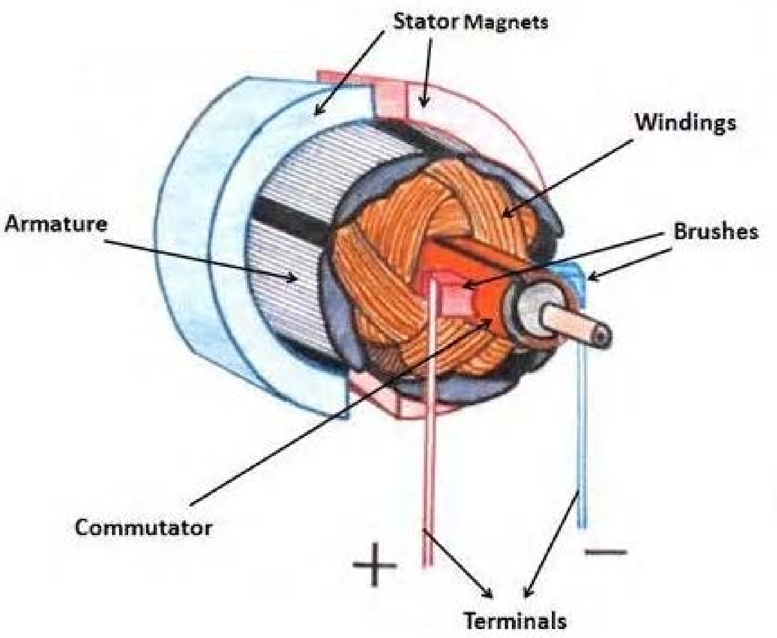


Figure 2.1: Components of a brushed motor

2.2 AC Motor

Unlike DC motors, this type of motor requires an alternating voltage supply to work. These motors usually have windings on the stator and permanent magnets on the rotor.
There are two categories that differ in number of phases:

- *Single-phase motors*: These motors are powered by a single-phase and starting from this one, a second phase is generated in quadrature.

- *Three-phase motor*: As the name implies, three-phase motors are powered by a three-phase system. Each phase is connected to at least one coil, so the number of windings must be a multiple of 3. Compared with single-phase motors, three-phase motors are very robust, relatively cheap and require little maintenance. They provide a steadier output and have uniform torque.

Another important difference is whether the motor is synchronized with the frequency of the supply voltage or not:

- *Synchronous motor*: in this motor the rotation speed is synchronized with the electric frequency. It means that the rotation frequency is the same or a multiple of the electric frequency. Synchronous motors are more efficient than asynchronous motors but they need an additional system to initially bring the rotor speed near to the target speed.

- *Asynchronous motor or Induction motor*: in this motor the angular speed of the rotor is lower than the rotation speed of the magnetic field generated by the windings of the stator. Asynchronous motors are simple and rugged in construction, they are robust and can operate in any environmental condition. Unlike synchronous motors, they have self starting torque and therefore they do not need to have external starting system.

2.3 Permanent Magnet Synchronous Motor (PMSM)

For this thesis, a three-phase AC synchronous brushless motor with permanent magnets will be used. The use of this motor is increasing fast in industrial environments, especially in small and medium power servo drives. They are essentially intended for high performance drives, where the particular specifications justify their cost which is usually high due to the presence of high quality permanent magnets.
In PMSM, the stator and the rotor are both shaped as a cylindrical crown of laminated ferromagnetic material and separated by an air gap. The rotor is composed by permanent magnets, while on the stator three-phase windings can be found.
Each phase has at least one coil that generates the imposed magnetic field and another coil on the other side, with the windings in the opposite direction, that generates the opposite magnetic field.
The number of coils of each phase is generally referred to as *poles.*
The term *Poles Pairs* is also used, meaning the pairs of poles for each phase.
The simplest PMSM has 6 coils, 2 for each phase, therefore in this configuration, the motor has 2 poles and 1 pole pairs. A simple scheme of this motor is reported in figure 2.2.
In this kind of motors, as already mentioned, the pattern of coils can be repeated, adding to each phase one or more pairs of coils. As a reference, in figure 2.3 it is shown a simple scheme of a 4 pair poles motor.
Finally, the windings of each phase are connected to pairs of terminals through which it is possible to supply power to the motor.
In this thesis, a tree-phase PWM inverter powered by a DC link has been selected as power source.

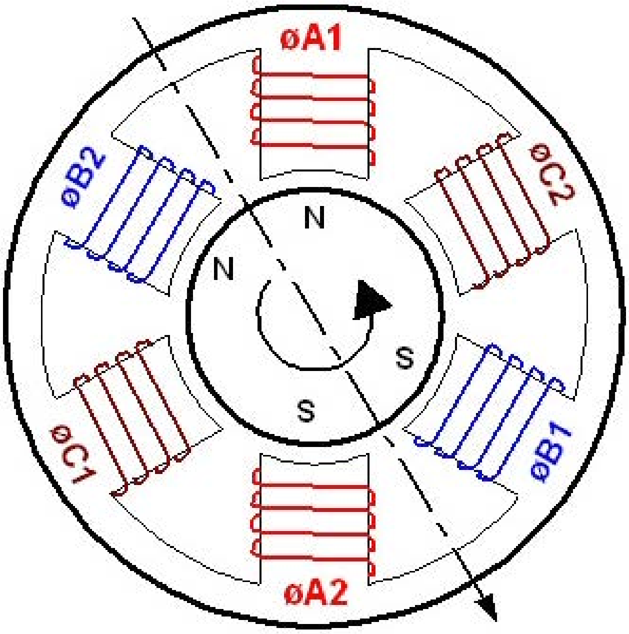


Figure 2.2: Scheme of a 1-Poles pair permanent magnets synchronous motor

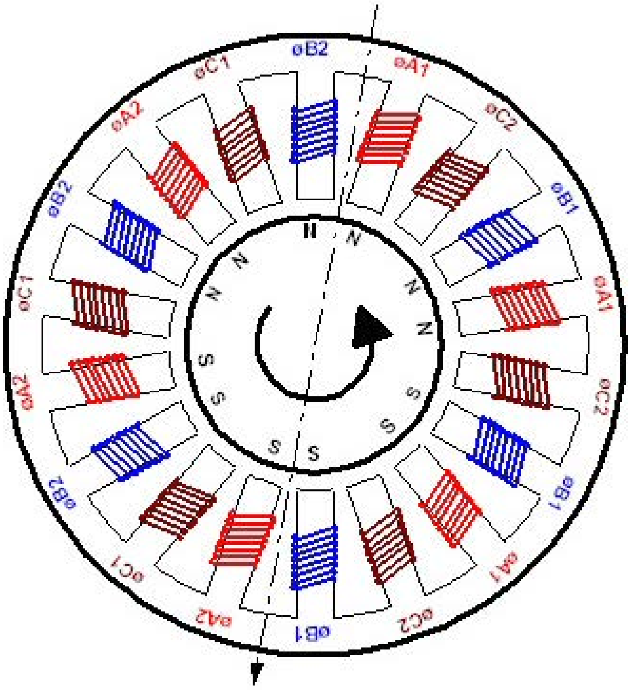


Figure 2.3: Scheme of a 3-Poles pairs permanent magnets synchronous motor

3 Controllers

In the development of a motor system, one of the modules that needs more attention in the design phase is the control module, also called the controller, since it is in this module that the signals that drive the system are generated.

In electric motors, the controller generates the voltage signals that drive the motor with the intention of bringing a specific system parameter to a certain target value.

The most common motor controls are the electronic controls that can be grouped into various categories based on their characteristics. For example, there are analog electronic controls and digital electronic controls, based on the hardware characteristics with which they are built.

Another important feature in which controls can be differentiated is based on whether or not a control uses the output signals to calculate the control signals. These controls are called respectively closed-loop controls and open-loop controls.

3.1 Open-loop Control

The open-loop control, also called non-feedback control, is independent from the output variable that it aims to control. In fact, it does not use feedback to determine if system output has achieved the desired target value[3], thus these kind of controller may be very inaccurate.

Open-loop systems are used when operation is fairly predictable and no particular accuracy of output values is required, for example, a cooling fan usually does not need precise speed control, and might simply be switched on and off.

In a DC motor, the simplest open loop control is based on voltage regulation since the speed is proportional to the applied voltage. Starting from this relationship, a voltage can be found to obtain a certain speed. However, if the motor load changes, or another external factor that affect the system changes, the voltage-speed relationship may change and therefore the motor speed associated with the current applied voltage. Since the control is open-loop there is no mechanism that corrects the applied voltage if not a human intervention.

For AC motor, one of the most famous control is Volt over Hertz control (V/Hz). In a synchronous motor voltage frequency is proportional to the speed, so varying the voltage frequency affects both the motor speed and the strength of the magnetic field: when the frequency is lowered (for achieving slower motor speed), the magnetic field increases and excessive heat is generated. On the other hand, when the frequency is increased (for achieving faster motor speed), the magnetic field decreases and lower torque is produced. In order to keep the magnetic flux constant, the Voltage over Hertz ratio must remain constant, this keeps torque production stable, regardless of the frequency. A schematic example of V/Hz control is shown in figure 3.1.

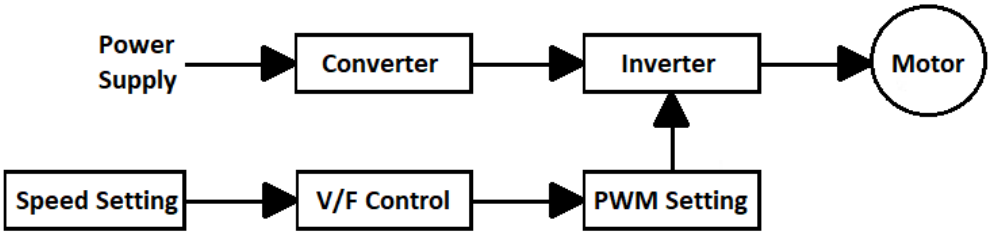


Figure 3.1: V/Hz control scheme

3.2 Closed-loop Control

A closed-loop control or feedback control system uses output parameters to adjust the input parameters, but its purpose is always to bring a quantity to a certain setpoint. In a motor control this quantity is usually the speed, therefore optical sensors and hall sensors can be used to read the rotor speed and position in order to provide feedback. Another possibility is to sense the back electromotive force, so using no sensor on the rotor shaft (sensorless control).
These kind of controllers are more expensive than open-loop systems, but guarantee continuous control over the quantity of interest, assuring also excellent precision and a fast response time to variations. In figure 3.2 it is shown a general block diagram of the closed-loop control.

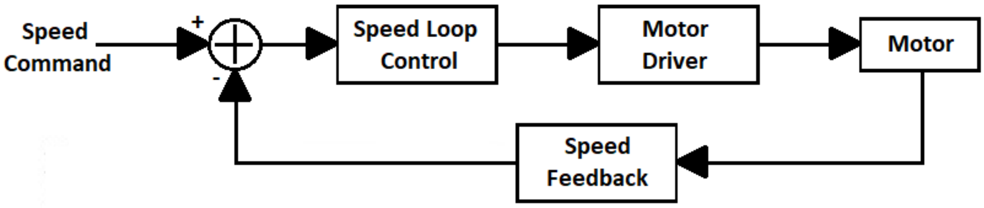


Figure 3.2: A simple closed-loop control

4 Field Oriented Control (FOC) for Three-phase motors

In this thesis a particular control is studied and implemented: the digital version of the field oriented control (FOC), used to drive a three-phase permanent magnet synchronous motor (PMSM).
FOC is a closed-loop control that uses motor current vector as feedback. In figure 4.1 it is shown this vector and its cartesian components, with cartesian axes related to the rotor position. The orthogonal component of the current with respect to the rotor is called I_q, while the parallel component is called I_d. Starting from the three-phase current, it is possible to obtain I_d and I_q using two transformations called Clarke transform and Park transform.

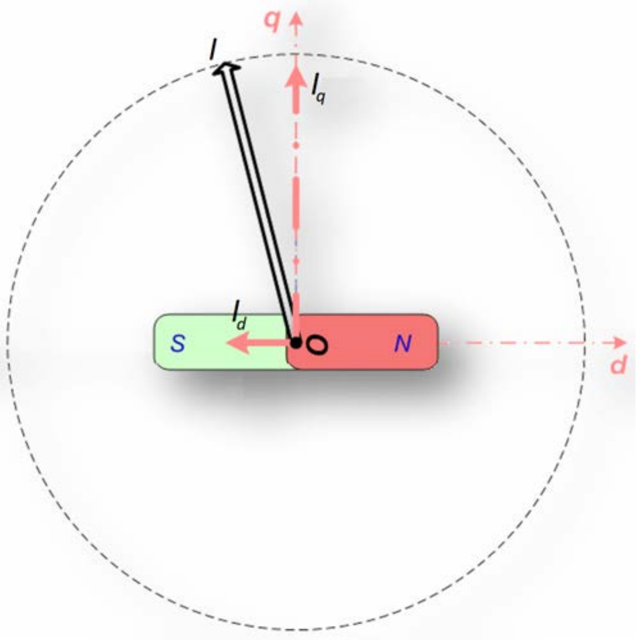


Figure 4.1: Example of Current Vector in a motor

The FOC controls I_q and I_d components to obtain the desired torque with the least possible currents, since the equation of the torque in a permanent magnet synchronous motor depends on I_q and it is:

$$T = \frac{3}{2} \cdot \frac{p}{2} \cdot \lambda_{pm} \cdot I_q$$

Where λ_{pm} is the flux linkage due to the permanent magnet and p is the number of poles.
For a motor control, the ideal situation is when the current vector is perpendicular to the rotor, thus having the component parallel to the rotor equal to zero. With these conditions, maximum torque is obtained and therefore maximum efficiency. With reference to figure 4.2, the I_d component must be 0 and I_q can be modulated to get the desired torque. Taking into account that in the synchronous motors the current vector rotates at the same speed as the rotor, the FOC uses these components because they are continuous and therefore easier to control than sinusoidal signals.

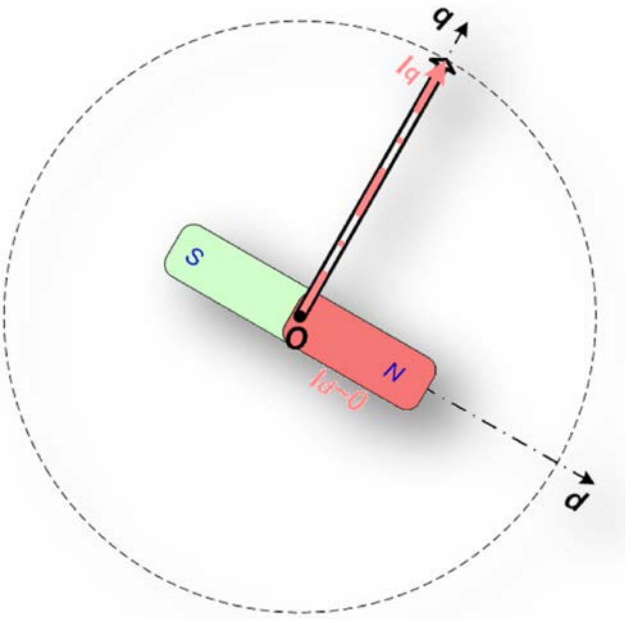


Figure 4.2: Ideal current components

Regarding the generation of the three-phase voltages signal that drive the motor, a space vector modulator (SVM) is used because it can easily generated signals with variable frequency and amplitude.
No rotor position is used so the controls use a module called observer that estimate

the rotor speed and the rotor position using the BEMF. There are more expensive solutions such as encoders or hall probes to estimate the rotor position but in this thesis they will not be taken into consideration. The control scheme in figure 4.3 is obtained combining the modules in a closed loop.

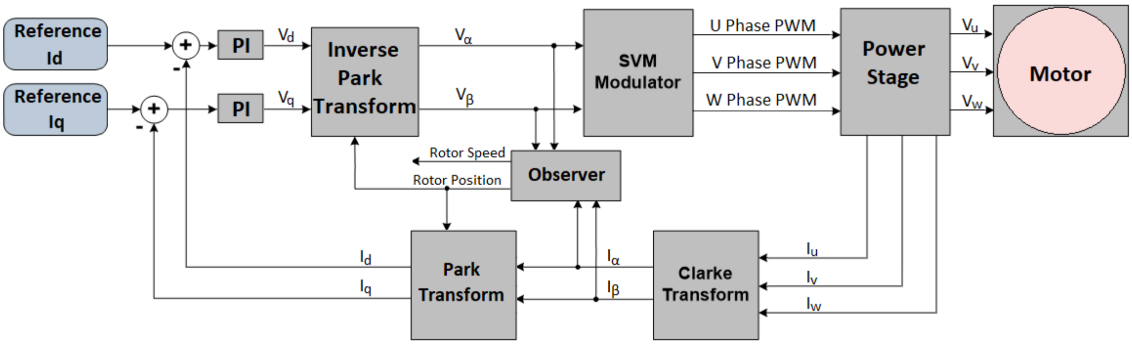


Figure 4.3: Theorical Scheme

All the blocks are connected to each other to form the feedback loops, the I_d loop and the I_q loop. Motor currents are read, transforms are applied and, once obtained, the I_d and I_q measured values are compared with the reference to obtain the error signals. Two Proportional-Integrative controls (PI) are added to correct feedback errors and bring the I_d and I_q equal to the references.
The PI outputs represent motor voltage vector in dq domain, used to calculate three-phase PWM signal via the space vector modulation.

4.1 Three-phase Vector Transformation

An ideal three-phase sinusoidal system consists of three sinusoidal signals with the same frequency and amplitude but with a phase difference of 120°. The three sinusoidal components V_u, V_v and V_w drawn on the plane and summed, give the rotating vector $\vec{v}$, as shown in figure 4.4.

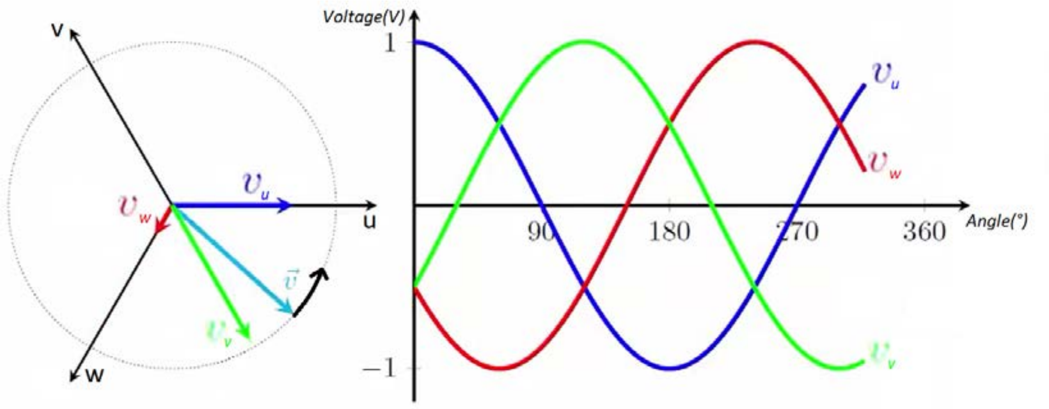


Figure 4.4: UVW threephase representation

4.1.1 Clarke Transform

If 3 components (the three-phases) are used to represent a vector (the rotating vector) in two dimensions plane there is redundancy. A two orthogonal component representation, called alpha-beta ($\alpha\beta$) transform and also known as the Clarke transform, is introduced to remove redundancy.
As shown in figure 4.5, the same rotating vector is now represented by only two orthogonal components and, as it can be noted, these components are still sinusoidal with a phase difference of 90°.

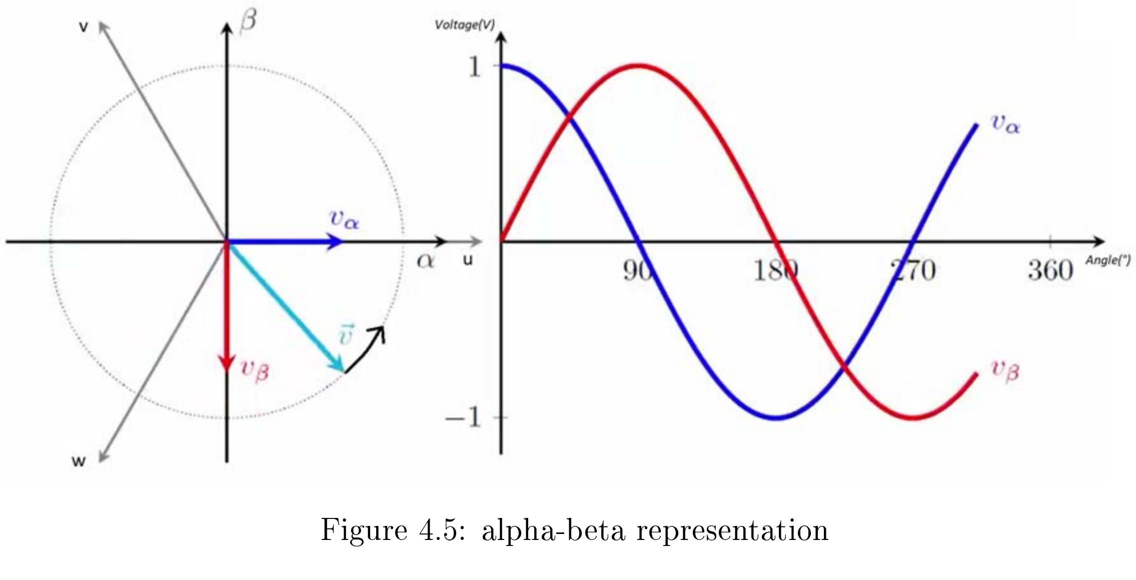


Figure 4.5: alpha-beta representation

If the three-phase vector components give zero sum, there is no loss of information and therefore the process is reversible. Otherwise, a third orthogonal component called gamma (γ) assumes a non-zero value. When working with motor currents, γ is considered equal to zero.
From a mathematical point of view, the Clarke transform can be seen as a multiplication between the three components U, V and W of a three-phase physical quantity and a transformation matrix.

$$\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix} = \frac{2}{3}\begin{bmatrix}1 & -\frac{1}{2} & -\frac{1}{2}\\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2}\\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix}\begin{bmatrix}U\\ V\\ W\end{bmatrix}$$

Where α,β and γ are the three components of a general three-phase physical quantity in $\alpha\beta$ domain.

4.1.2 Park Transform

In synchronous motor, voltage rotating vector, current rotating vector, magnetic field and rotor have the same rotating speed. Starting from this assumption and turning the orthogonal reference of an angle equal to the angular position of the rotor at that given moment, two continuous components are obtained, as shown in figure 4.6. A third component called zero (z) is present if gamma component of Clarke transform isn't equal to zero. This further transformation is called dq0 transform or Park transform.

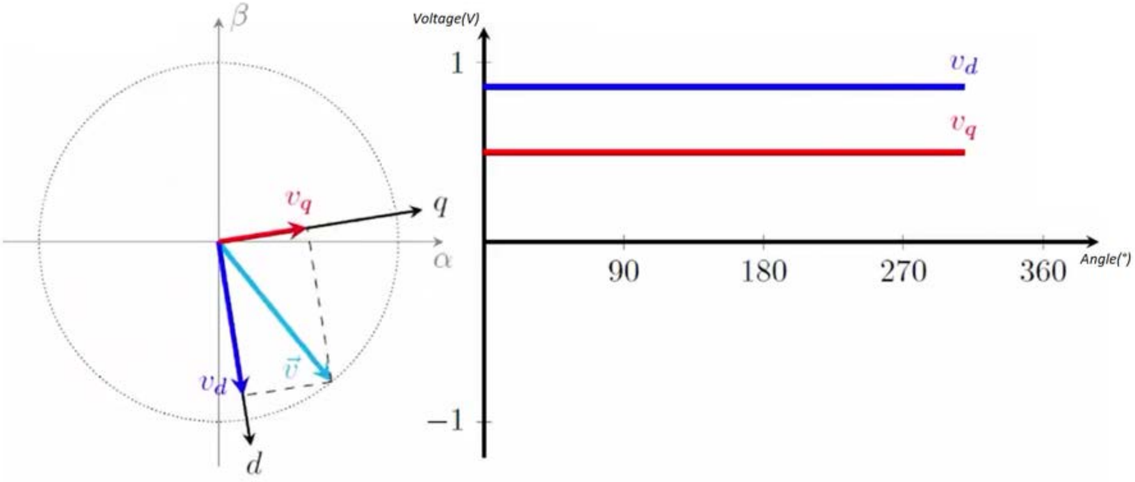


Figure 4.6: d-q representation

From a mathematical point of view, the Park transform can be seen as a multiplication between the three components α, β and γ of a three-phase physical quantity and a transformation matrix.

$$\begin{bmatrix} d \\ q \\ z \end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

Where d,q and z are the three components of a general three-phase physical quantity in dq domain.

4.2 Rotor position estimation

In the previous chapter it is shown that it's necessary to know the position of the rotor to apply the Park transform.
As mentioned before, it was decided to implement this control without sensors on the motor, but estimating the position (and consequently the speed) using the inverse electromotive force caused by the rotor magnetic field. It will therefore be necessary to access a current measurement. In this thesis other restrictions are imposed: it will be necessary to access the current of each phase in a synchronized way and be able to sample it at every cycle. The sample must also be as recent as possible.
The module where the position of the rotor is estimated from the current is generally called observer. The functioning of the observer is based on the phase-voltage equation of the three-phase PMSM:

$$v_u = R_u \cdot I_u + L_u \frac{di_u}{dt} + e_u \tag{4.1}$$

$$v_v = R_v \cdot I_v + L_v \frac{di_v}{dt} + e_v \tag{4.2}$$

$$v_w = R_w \cdot I_w + L_w \frac{di_w}{dt} + e_w \tag{4.3}$$

where v_u, v_v, v_w are phase stator voltages, R_u, R_v, R_w are stator winding resistance, i_u, i_v, i_w are phase stator current, L_u, L_v, L_w are phase stator inductance, and e_u, e_v, e_w are phase back-EMF. The inverse electromotive force depends on the rotor speed and its position, for each phase the formulas are[18]:

$$e_u = k_e \omega_r sin(\theta_r) \tag{4.4}$$

$$e_v = k_e \omega_r sin(\theta_r - 120) \tag{4.5}$$

$$e_w = k_e \omega_r sin(\theta_r + 120) \tag{4.6}$$

Where ω_{ris} is the rotor electrical speed and θ_{ris} is the rotor electrical position. As it can be noted, this equations contains a reference to the rotor speed and it's where the observer measures its value. The observer used in this thesis is patented thus further details can not be provided.

4.3 Three-phase voltage generation (SVM)

In this type of control the space vector modulation (SVM) is used to generate the PWM three-phase signal. These PWM signals are used by the power stage to drive the motor. SVM consists of 8 vectors and each vector is characterized by 3 numerical components that indicate the logical state of each pwm phase, as shown in table 4.1.

Vector	**U phase PWM**	**V phase PWM**	**W phase PWM**
[0,0,0]	Low	Low	Low
[1,0,0]	High	Low	Low
[1,1,0]	High	High	Low
[0,1,0]	Low	High	Low
[0,1,1]	Low	High	High
[0,0,1]	Low	Low	High
[1,0,1]	High	Low	High
[1,1,1]	High	High	High

Table 4.1: SVM Vectors

If the thee vector components are considered as phase U, phase V and phase W components and the Clarke transform is applied, 8 vectors in the $\alpha\beta$ plane are obtained, as shown in figure 4.7. As it can be noted, the vectors [0,0,0] and [1,1,1] don't bring AC component and therefore these vectors have length equals to 0.

Using 2 of the 8 vectors with appropriate times, all vectors present in the hexagon area of the figure can be generated. As it can be noted, the vectors are normalized to V_{max} so that 1 represents the voltage of V_{max}, equal to the high state voltage of PWM.

In figure 4.8 an example is shown: for generating V_{sp}, a possible combination is $V2[1,1,0]$ for a time equal to the length of $T[1,1,0]$ and $V1[1,1,0]$ for a time equal to the length of $T[1,0,0]$.

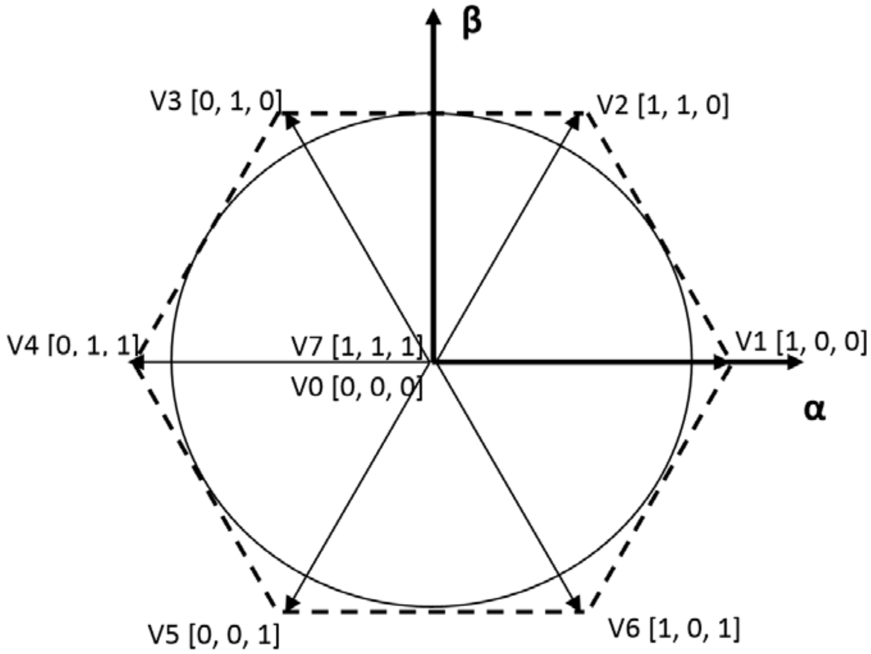


Figure 4.7: Space Vector Modulation Hexagon

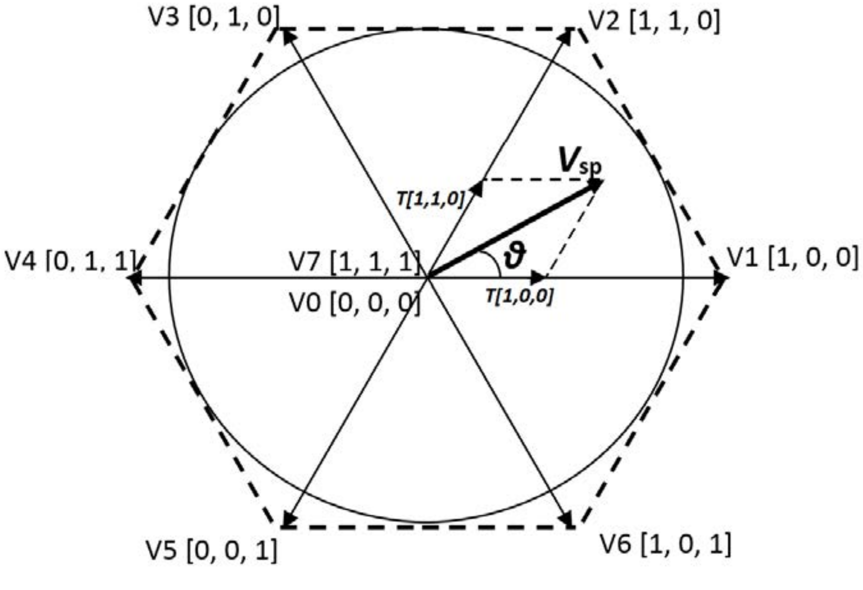


Figure 4.8: Space Vector Modulation Time example

As it can be noted, the time is normalized to duty cycle period, therefore the sum

must be 1. In the rest of the time, [1,1,1] vector or [0,0,0] vector (null vectors) can be generated. Another possibility to get a zero vector is to generate two opposite active vectors for the same time so that they eliminate each other and therefore they don't bring AC contribution.
The PWM that generates the vector V_{sp} in figure 4.8 is shown in figure 4.9.

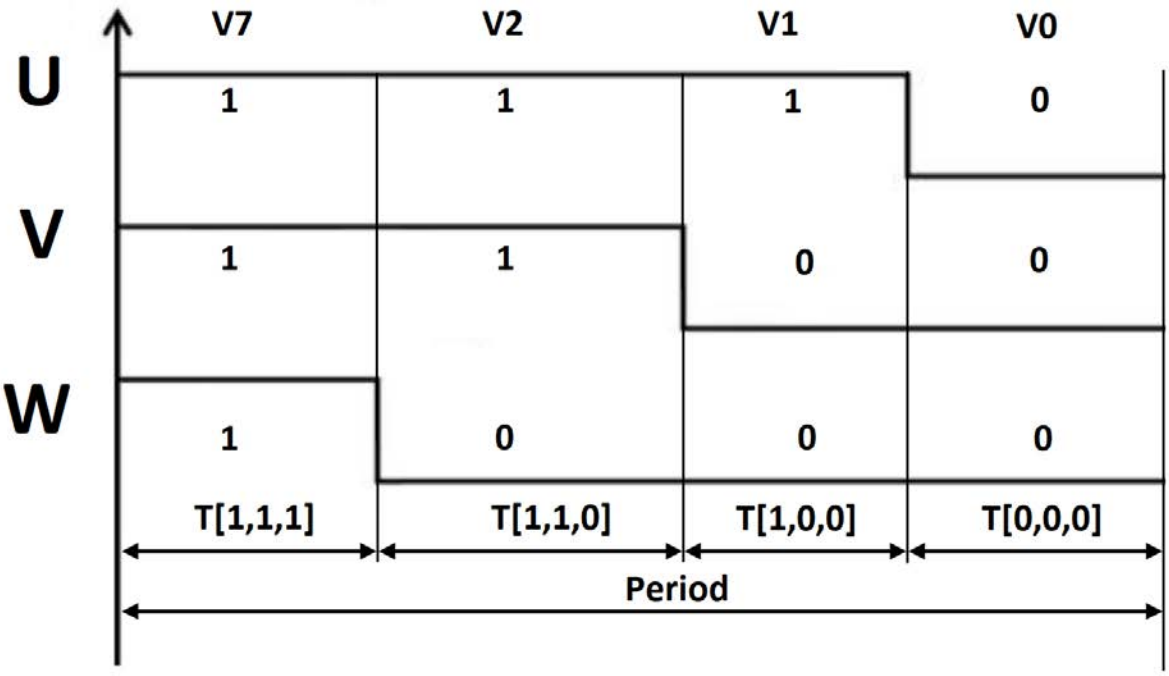


Figure 4.9: PWM example

4.4 Field Oriented Control Implementation

To implement the FOC controller it is necessary to use a microcontroller that meets the following specifications:

- Fast ADCs for the measurement of the phase currents
- Modules that generate PWM signal
- Mathematical unit for data processing
- Module that permits synchronism

The selected model is the Infineon XMC4400, which reflects all the features required. The power stage is represented by a 3 half-bridges with the respective drivers connected to the motor and commanded by PWM signals. The half bridges have in each

leg a resistance necessary to measure the current from which the position and speed of the rotor will be estimated (three shunt). The current measurement can also be made on a single resistance connected in common to the three legs (single shunt), but this method will be not studied in this thesis.
A graphic representation of the used hardware is shown in figure 4.10.
The software is developed in C.

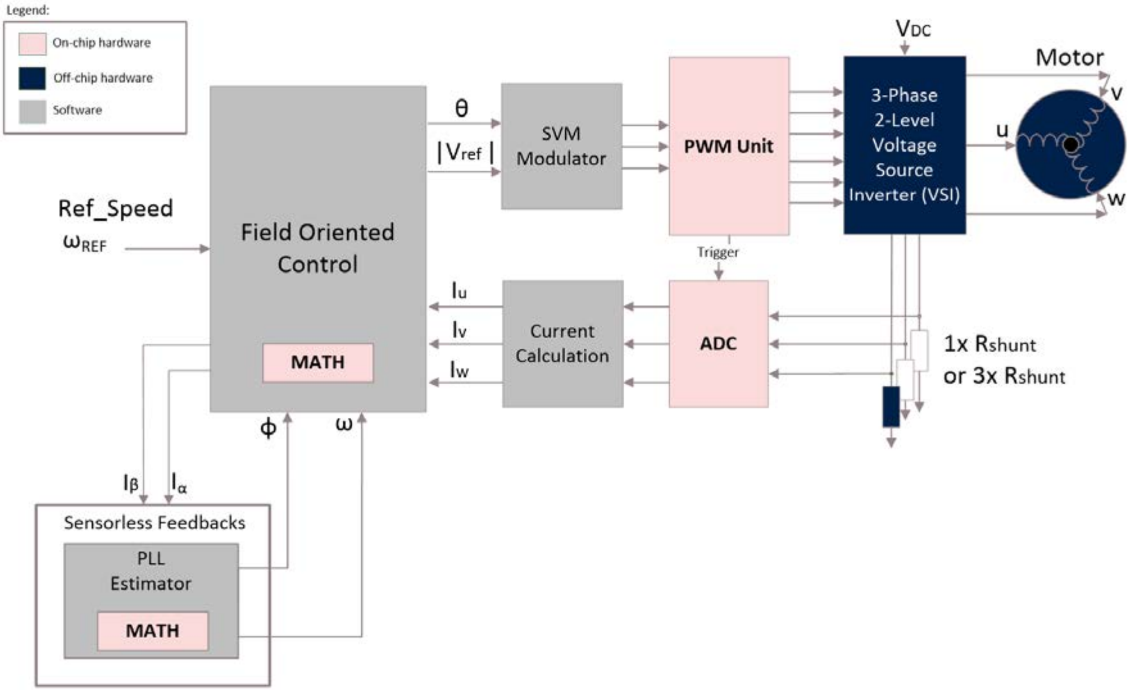


Figure 4.10: Foc Hardware scheme

5 Hardware

5.1 The XMC4400 Microcontroller

The microcontroller used in this thesis is the Infineon XMC4400 from XMC4000 family based on ARM-M4. Other microcontrollers that have the necessary features are the XMC1300 and XMC1400 from XMC1000 family based on ARM-M0. In fact all these microcontrollers have the necessary features, as show in Table 5.1.

Feature	**XMC1300**	**XMC1400**	**XMC4400**
Architecture	ARM-M0	ARM-M0	ARM-M4
Microcontroller Clock	32MHz	48MHz	120Mhz
Peripheral Clock	64MHz	96MHz	120Mhz
Math Unit	CORDIC	CORDIC	FPU

Table 5.1: Microcontrollers features

Where CORDIC is a coprocessor that implements CORDIC algorithm.

5.1.1 CPU

The XMC4400 features the ARM Cortex-M4 processor, a high performance 32-bit processor designed for the microcontroller market. This CPU offers significant benefits to users, including:

- Outstanding processing performance combined with fast interrupt handling
- Enhanced system debug with extensive breakpoint and trace capabilities
- Platform security robustness, with integrated memory protection unit (MPU)
- Ultra-low power consumption with integrated sleep modes
- IEEE754-compliant single-precision FPU

- Power control optimization of system components
- Deterministic, high-performance interrupt handling for time-critical applications

The Cortex-M4 processor is built on a high-performance processor core, with a 3-stage pipeline Harvard architecture, making it ideal for demanding embedded applications. The processor delivers exceptional power efficiency through an efficient instruction set and extensively optimized design, providing high-end processing hardware including IEEE754-compliant single-precision floating-point computation, a range of single-cycle and SIMD multiplication and multiply-with-accumulate capabilities, saturating arithmetic and dedicated hardware division.
The instruction set provides the exceptional performance expected of a modern 32-bit architecture, with the high code density of 8-bit and 16-bit microcontrollers. The Cortex-M4 processor closely integrates a configurable Nested Vectored Interrupt Controller NVIC, to deliver industry leading interrupt performance. It includes a non-maskable interrupt (NMI), and provides up to 64 interrupt priority levels and 112 interrupts.
To optimize low-power designs, the NVIC integrates with the sleep modes, that include a deep sleep function that enables the entire device to be rapidly powered down while still retaining program state. For reference, in figure 5.1 CPU block diagram is shown.

The XMC4400 has many digital port pins which can be used as General Purpose I/Os (GPIO) and are connected to the on-chip peripheral units. The PORTS provide a generic and flexible software and hardware interface for all standard digital I/Os. Each port slice has the same software interfaces for the operation as General Purpose I/O and it further provides the connectivity to the on-chip periphery and the control for the pad characteristics.

The Cortex-M4 implements a FPU with the FPv4-SP floating-point extension. It fully supports single-precision add, subtract, multiply, divide, multiply and accumulate, and square root operations. It also provides conversions between fixedpoint and floating-point data formats, and floating-point constant instructions. The FPU contains 32 single-precision extension registers, which can be also accessed as 16 doubleword registers for load, store, and move operations.

In this thesis, the NVIC is used to handle cycle machine interrupts while the FPU is used as a math unit to perform complex operations such as trigonometry.

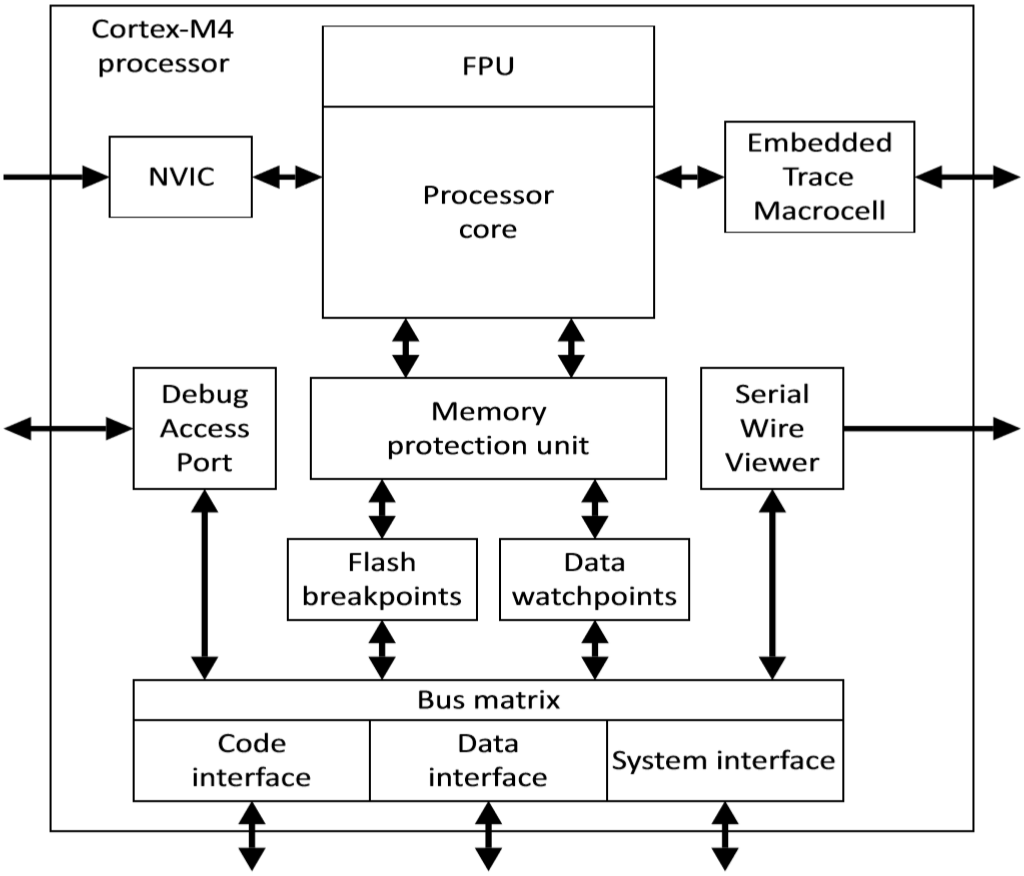


Figure 5.1: CPU Block Diagram[13]

5.1.2 VADC

To read external analog signals, the XMC4400 features a Versatile Analog to Digital Converter that is comprised of 4 independent analog to digital converters, each capable of converting with a resolution of 12-bits at 2 MSamples/sec. This enables highly accurate signal measurement for currents, voltages, temperature signals. It provides a series of analog input channels connected to a cluster of Analog/Digital Converters using the Successive Approximation Register (SAR) principle to convert analog input values (voltages) to discrete digital values. Each ADC features a total of 8 input channels.
For reference, in figure 5.2 and 5.3 VADC structure and single ADC structure are shown.

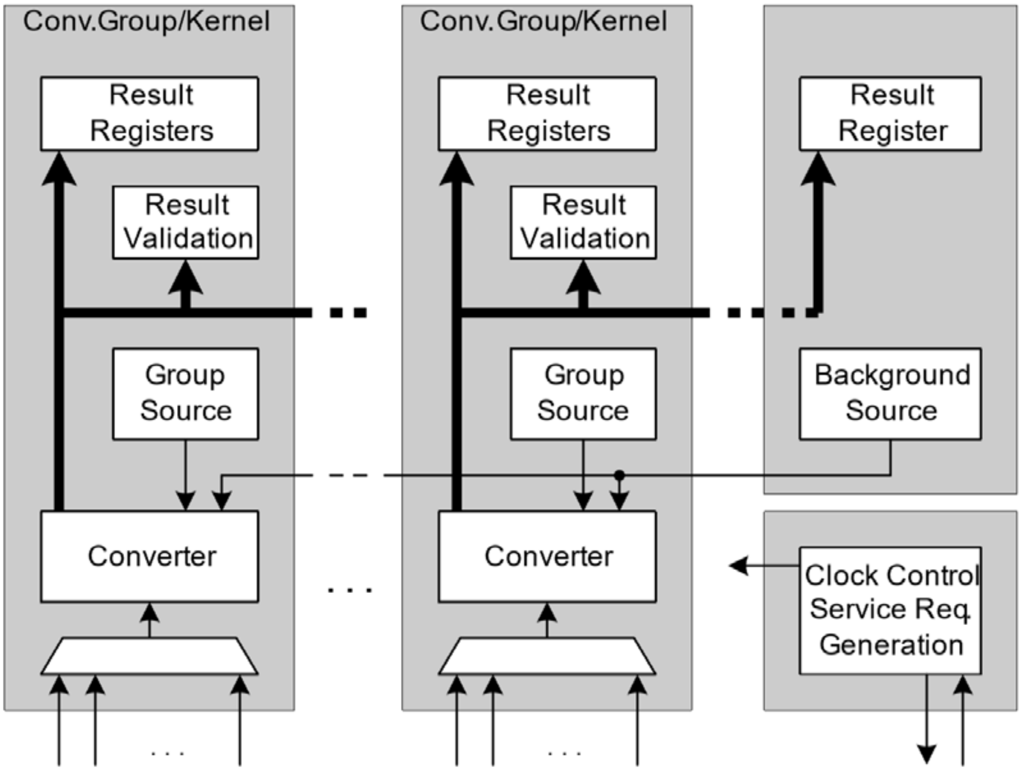


Figure 5.2: VADC Cluster Structure[13]

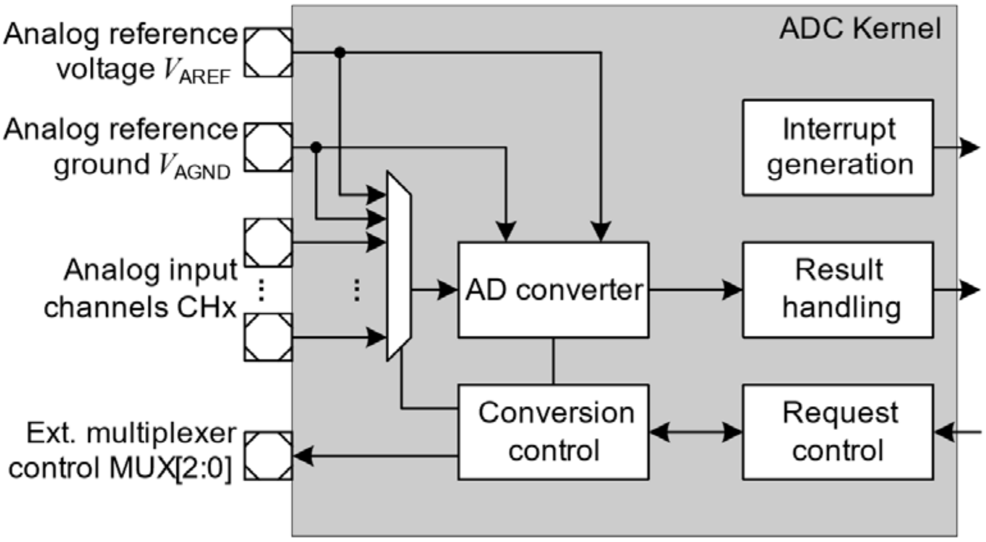

ADC Kernel
Analog reference voltage V_{AREF}
Analog reference ground V_{AGND}
Analog input channels CHx
Ext. multiplexer control MUX[2:0]
AD converter
Conversion control
Interrupt generation
Result handling
Request control

Figure 5.3: Single ADC Structure[13]

Each converter of the ADC cluster can operate independent of the others, controlled by a dedicated set of registers and triggered by a dedicated group request source. The results of each channel can be stored in a dedicated channel-specific result register or in a group-specific result register.
The request sources are of 3 types: Queue, Scan and Background, which differ from one another for the complexity of the sampling sequence. The Queue sources and Scan sources are called *groups* request source. In this thesis only Queue source and Background source are used.
A background request source can access all analog input channels that are not assigned to any group request source. These conversions are executed with low priority. The background request source can, therefore, be regarded as an additional background converter.
Because all request sources can be enabled at the same time, an arbiter resolves concurrent conversion requests from different sources. Certain channels can also be synchronized with other ADC kernels, so several signals can be converted in parallel. The ADC cluster features many functionality, such as:

- Input voltage range from 0 V up to analog supply voltage
- External analog multiplexer control, including adjusted sample time and scan support
- Conversion speed and sample time adjustable to adapt to sensors and reference
- Programmable arbitrary conversion sequence (single or repeated)
- Conversions triggered by software, timer events, or external events
- Selectable result width of 8/10/12 bits
- Configurable limit checking against programmable border values
- Effective result handling for bursts of high-speed conversions
- Synchronous sampling of up to 2 or 4 input signals

Conversion parameters, such as sample phase duration, reference voltage, or result resolution can be configured for 4 input classes (2 group-specific classes, 2 global classes) and each channel can be individually assigned to one of these input classes. The input channels can, thus, be adjusted to the type of sensor (or other analog sources) connected to the ADC.

Several independent ADC implemented in the XMC can be synchronized for simultaneous measurements of analog input channels. The synchronization mechanism for parallel conversions ensures that the sample phases of the related channels start simultaneously. Synchronized ADCs convert the same channel that is requested by the master, if the channels to be sampled are different, a VADC function called *alias* can be used. This function consists in redirecting a certain channel to another one, therefore non-equal channels can be sampled synchronously.

Different values for the resolution and the sample phase length of each ADC for a parallel conversion are supported.
During Synchronized Conversions, one ADC operates as synchronization master, the other ADC(s) operate(s) as synchronization slave(s).

Thanks to all these features, the VADC is used in this thesis to sample motor phase currents in a synchronous way using a queue source, while other less important measurements such as DC link voltage are in background source.

5.1.3 Capture/Compare Unit 8 (CCU8) (for PWM generation)

The CCU8 peripheral functions play a major role in applications that need complex Pulse Width Modulation (PWM) signal generation, with complementary high side and low side switches, multi phase control or output parity checking.
These functions in conjunction with a very flexible and programmable signal conditioning scheme, make the CCU8 the must have peripheral for state of the art motor control, multi phase and multi level power electronics systems.
The internal modularity of CCU8, translates into a software friendly system for fast code development and portability between applications.
The CCU8 unit is comprised of four identical 16 bit Capture/Compare Timer slices and each Timer Slice can work in Compare or in Capture Mode. In Compare Mode, one has two dedicated compare channels that enable the generation of up to 4 PWM signals per Timer Slice (up to 16 PWM outputs per CCU8 unit), with dead time insertion to prevent short circuits in the switches, while in Capture Mode a set of up to four capture registers is available. In this thesis, the CCU8 module has been used only in compare mode.
Each CCU8 module has four service request lines that can be easily programmed to act as synchronized triggers between the PWM signal generation and an ADC conversion.
Some of the general Features are:

- 16 bit timer cells
- Programmable low pass filter for the inputs
- Programmable clock prescaler
- Normal timer mode
- Three counting schemes – center aligned – edge aligned – single shot
- Asymmetric PWM generation
- TRAP function
- Dead time generation

In figure 5.4, it has been reported a general scheme of the CCU8 for completeness.

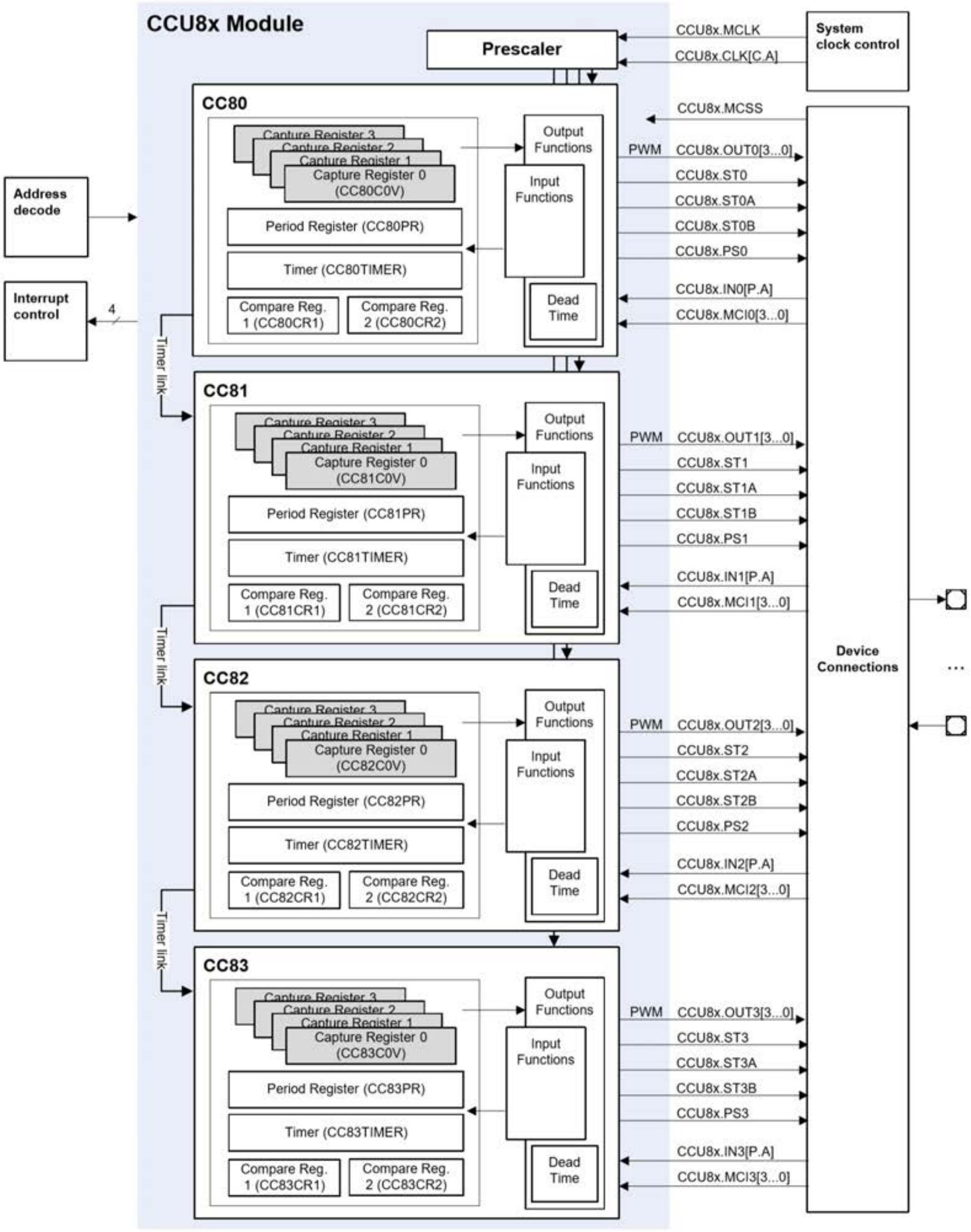


Figure 5.4: CCU8 Diagram Block[13]

Each CCU8 timer slice provides an associated shadow register for the period and the two compare values. This facilitates a concurrent update by software for these three parameters, with the objective of modifying during run time the PWM signal period and duty cycle.

Each CCU8 slice timer can be programmed into three different counting schemes:

- Edge aligned
- Center aligned
- Single shot (edge or center aligned)

These three counting schemes can be used as stand alone without the need of selecting any inputs as external event sources, nevertheless it is also possible to control the counting operation via external events like, timer gating, counting trigger, external stop, external start, etc. This enables a cycle by cycle update of the PWM frequency and duty cycle.
In this thesis, *Asymmetrical Edge Aligned* mode is used to generate the PWM signals. This mode consists in using the two channels combined to generate an asymmetric PWM output. Each single channel is in *Edge aligned* mode and the active status of the asymmetric PWM is set when a compare match with the compare value of channel 1 occurs and is cleared when a compare match with the compare value of channel 2 occurs.
In other words, the compare match of channel 1 represents the rising edge of the symmetric edge align PWM while the compare match of channel 2 represents the falling edge.

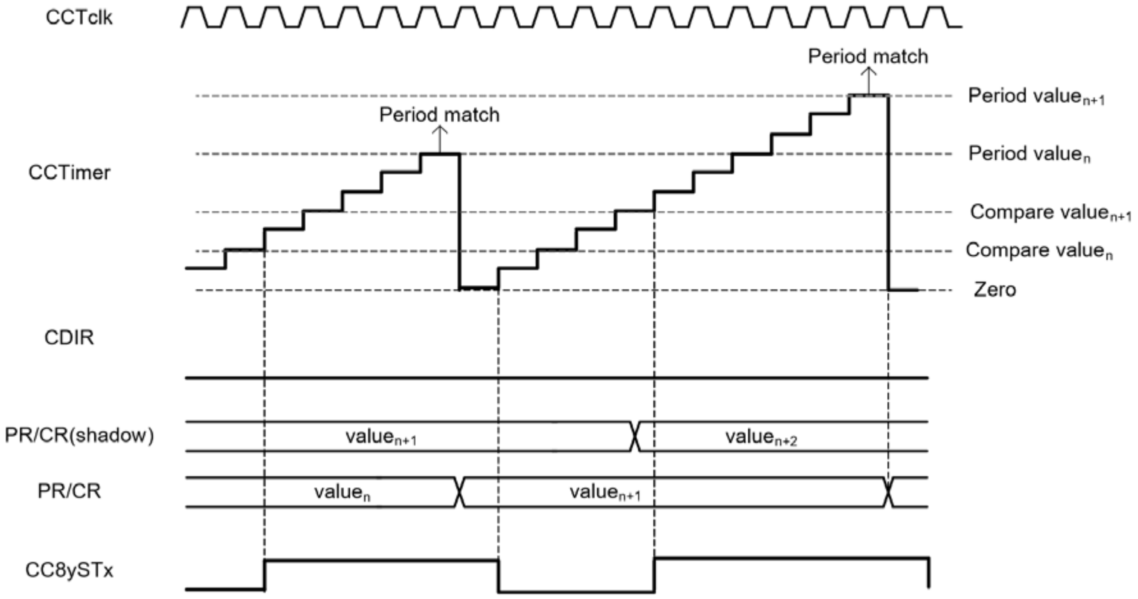


Figure 5.5: Edge Aligned[13]

Edge aligned mode is the default counting scheme. In this mode, the timer is incremented until it matches the value programmed in the period register and when period match is detected the timer is cleared to 0x0000 and continues to be incremented, as shown in figure 5.5. In this mode, the value of the period register and compare registers are updated with the values written by software into the correspondent shadow register, every time that an overflow occurs (period match). In CCU8 there is also a TRAP functionality that allows the PWM outputs to react on the state of an input pin. This functionality can be used to switch off the power devices if the TRAP input becomes active, in motor control it is generally connected to a special pin of the inverter. By doing so in case of overload, the CCU8 turns off and the control goes in a safe mode.

In this thesis, the CCU8 is used to generate PWM signals, synchronism signals and ADC trig signals. Also trap function is used for safety reasons.

5.1.4 Capture/Compare Unit 4 (CCU4) (for secondary timer)

The CCU4 peripheral is a major component for systems that need general purpose timers for signal monitoring/conditioning. The internal modularity of CCU4, translates into a software friendly system for fast code development and portability between applications. Similarly to CCU8, in CCU4 each module is comprised of four identical 16 bit Capture/Compare Timer slices and each timer slice can work in compare mode or in capture mode. In compare mode one compare channel is available while in capture mode, up to four capture registers can be used in parallel. Each CCU4 module has four service request lines and each timer slice contains a dedicated output signal, enabling the generation of up to four independent PWM signals. Besides, all four CCU4 timer slices, are identical in terms of available functions and operating modes and a built-in link between the four timer slices is also available, enabling a simplified timer concatenation and sequential operations.

The CCU4 module has peculiar properties and features similar to those of the CCU8, but in this thesis they are not used.
Although therefore it could be used to generate PWM, this unit is only used to time a slow machine cycle, with a simple period match interrupt.

5.1.5 Window Watchdog Timer (WDT)

Purpose of the Window Watchdog Timer module is improvement of system integrity. WDT triggers the system reset or other corrective action like e.g. non-maskable interrupt if the main program, due to some fault condition, neglects to regularly service the watchdog. The intention is to bring the system back from unresponsive state into normal operation. The WDT timer is a 32-bit counter, which counts up from 0H. It can be serviced while the counter value is within the window boundary, i.e. between the lower and the upper boundary value. Correct servicing results in a reset of the counter to 0H.

In this thesis the watchdog is used for safety reasons and it is fed in a cycle machine.

5.2 Inverter

Another main components used in motor control is the inverter which is connected to the microcontroller board. In this thesis, the inverter that has been used is a three-phase inverter, included in the board Infineon MOT_GPDLV-V2 Driver board. In figure 5.6 a general scheme of MOT_GPDLV-V2 board is shown.

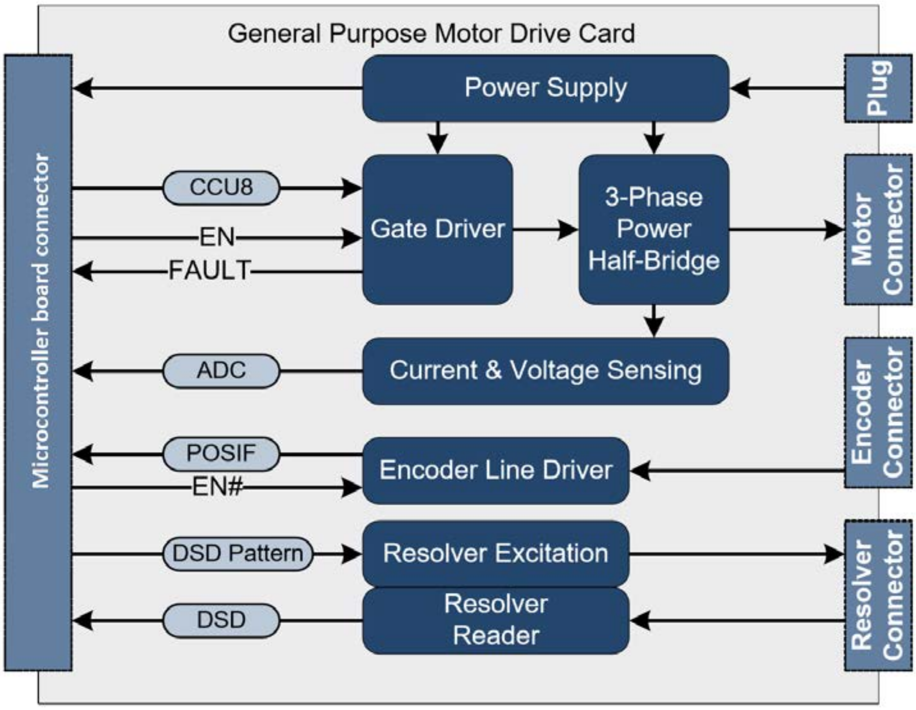


Figure 5.6: Inverter Blocks

As already mentioned, the position resolver is not used and neither is the encoder driver. Instead, the most important blocks in the inverter board are the gate driver,

the 3-phase power half-bridge and the current/voltage sensing module.
The brushless PMSM motor is connected directly to the board via a dedicated connector. In figure 5.7 it is shown the MOT_GPDLV-V2 circuit.
The inverter mainly consists of a *three-leg semi-bridge*, where the motor is connected. The half-bridge switches are controlled by a circuit called the *gate driver*. To control the inverter, the microcontroller provides 3 PWM signals and their negatives versions which command the gate driver.
In each leg low-side there is a shunt resistor: flowing through it, the current of each motor phase generates a voltage drop at its ends. This voltage is read by the microcontroller and therefore the program can calculate the current value starting from the voltage value and the resistance value. Also a common shunt resistor is present to measure the common current.
Other important signals present in the inverter are:

- FAULT: when an overcurrent occurs, this signal is in an active state.
- ENPOW: it is an enable line command for the gate driver.

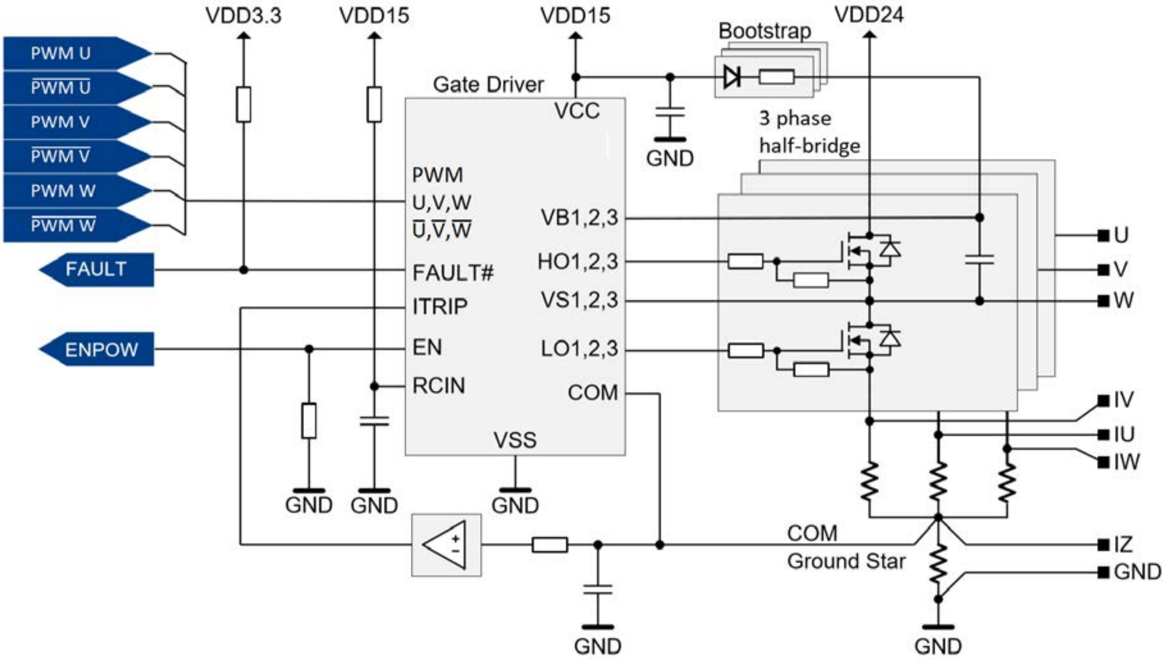


Figure 5.7: Inverter example circuit

5.2.1 Hardware Current scaling

Between leg resistors and ADC there is an amplifier, configured as shown in 5.8. An offset (OFFS) is added to each leg resistors voltage because the ADC can not read negative values. This offset, generally half of the maximum ADC voltage, is generated via a voltage divider or a DAC. The final value is then amplified by the amplifier and it is ready to be read by the ADC.

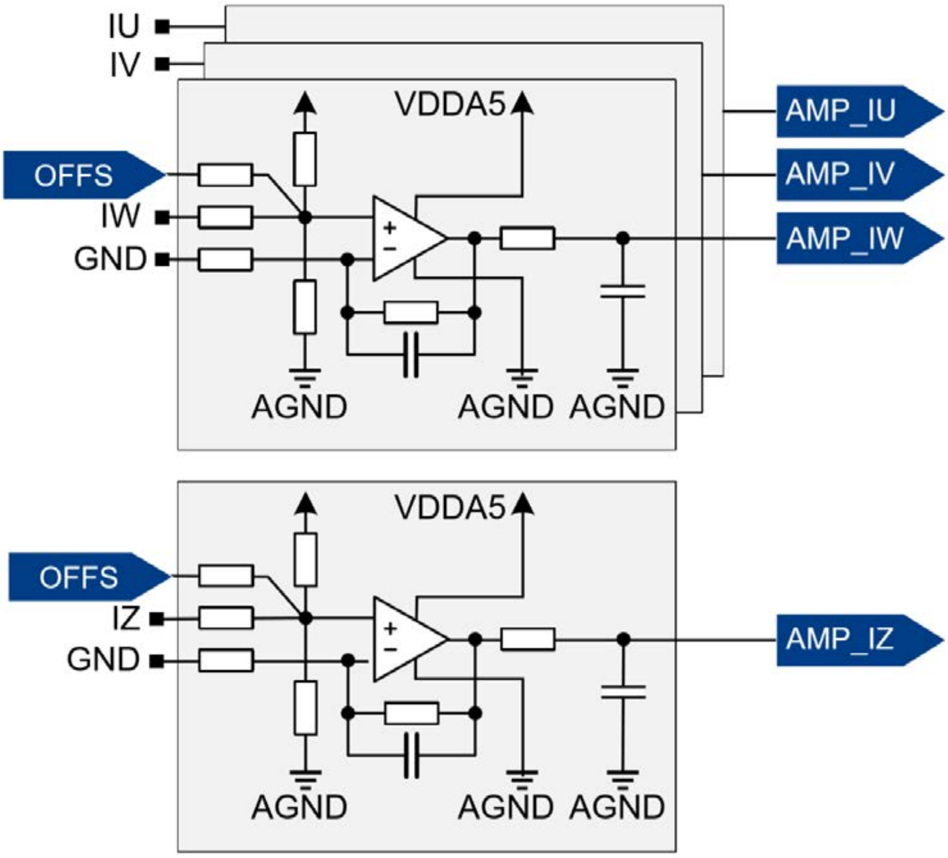


Figure 5.8: Current Scaling circuit

6 Software

The field oriented control algorithm is implemented by the software, which is also in charge of rotor speed and position estimation, hardware configuration and interrupt routines service.

6.1 Implemented control scheme

In figure 6.1 it's reported the general control scheme implemented.
The aim of the PMSM FOC motor control is to offer functionality to drive the PMSM motors with sensorless modules.
The phase currents are measured by the ADC and converted into a dq0 domain (section 4.1), the resulting components are then used as a feedback for the PI controller. From the PI's outputs, the module and phase are calculated to guide the PWM. The PMSM FOC software provides high level of configurability and modularity to address different motor control applications.

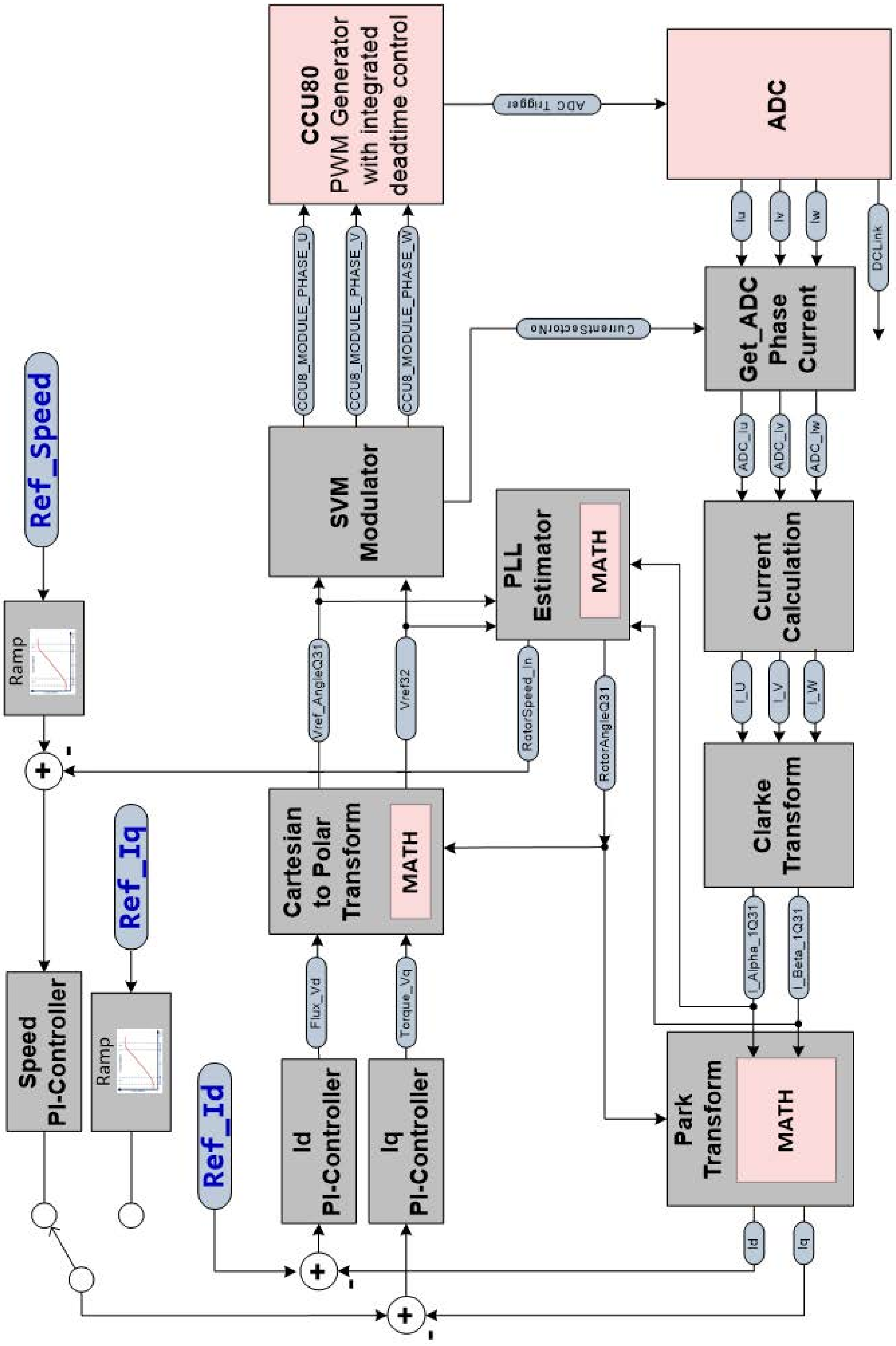


Figure 6.1: FOC Scheme

6.1.1 Difference from theoretical scheme

In the implemented scheme, the inverse Park transformation has been substituted with a cartesian to polar transformation and therefore, the SVM modulator has been modified to process the proper inputs. Thanks to this substitution, it is possible to achieve higher speed in terms of execution times.

Furthermore, as it can be noted in figure 6.1, it has been added a second feedback loop for regulating the rotor speed. The loop is closed on the rotor estimated speed and is used as a feedback for a PI controller that regulates the I_q current.

Another choice worthy of note is that the generation of the PWM signal is made via hardware, starting from the duty cycle values calculated by the space vector modulator.

6.1.2 Configurations of the control scheme

Five types of schemes have been implemented based on the number of the modules and interconnections:

Open-Loop Voltage Control

This is the V/Hz control, therefore it has no feedback loop. In the V/Hz control the ratio between voltage and rpm is proportional (or linearly correlated). In figure 6.2, it is shown this link and the various configurations that can be edited in the firmware.

In figure 6.3, it is shown the resulting control scheme. The reference speed is increased via a ramp to reach the desired target. The voltage module is obtained starting from the reference speed using the V/Hz multiplicative constant, while the voltage angle is increased proportionally to the reference speed to keep the rotor at the desired RPM.

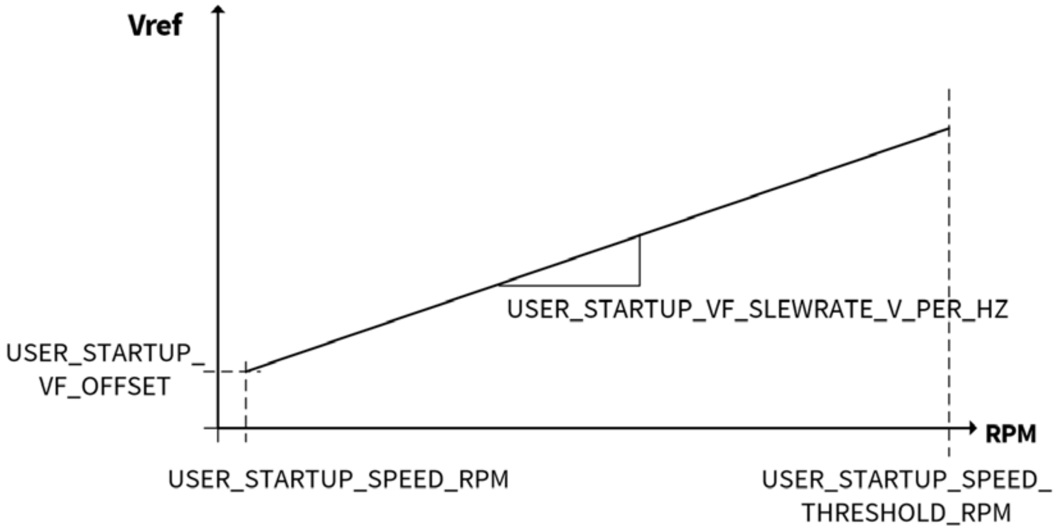


Figure 6.2: Voltage/rpm graph

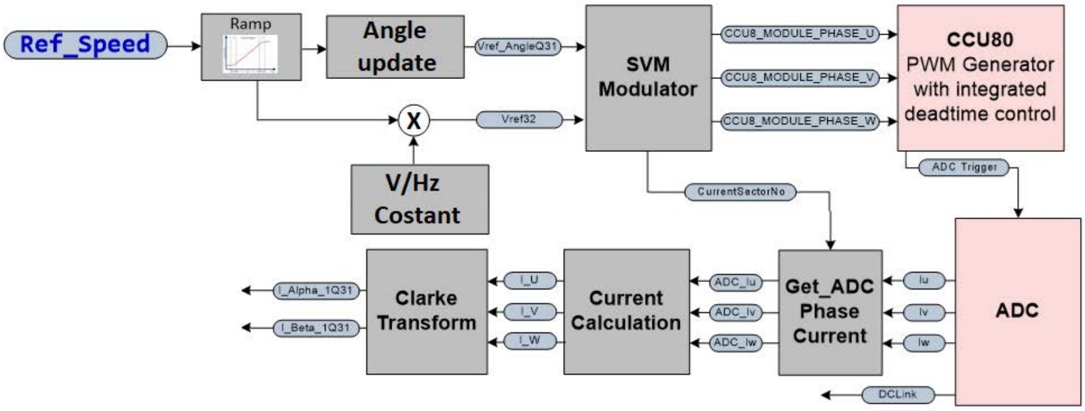


Figure 6.3: V/Hz Scheme

Speed Control Direct FOC Startup

With this scheme, the motor is started directly by the speed field oriented control, thus starting auxiliary systems are not needed. As it can be noted in figure 6.3, in this configuration there are 2 feedback loop. The I_d current component is controlled with reference equal to 0, as explained in chapter 4, while the I_q component is controlled by a reference torque (first loop) calculated starting from a speed loop (second loop).

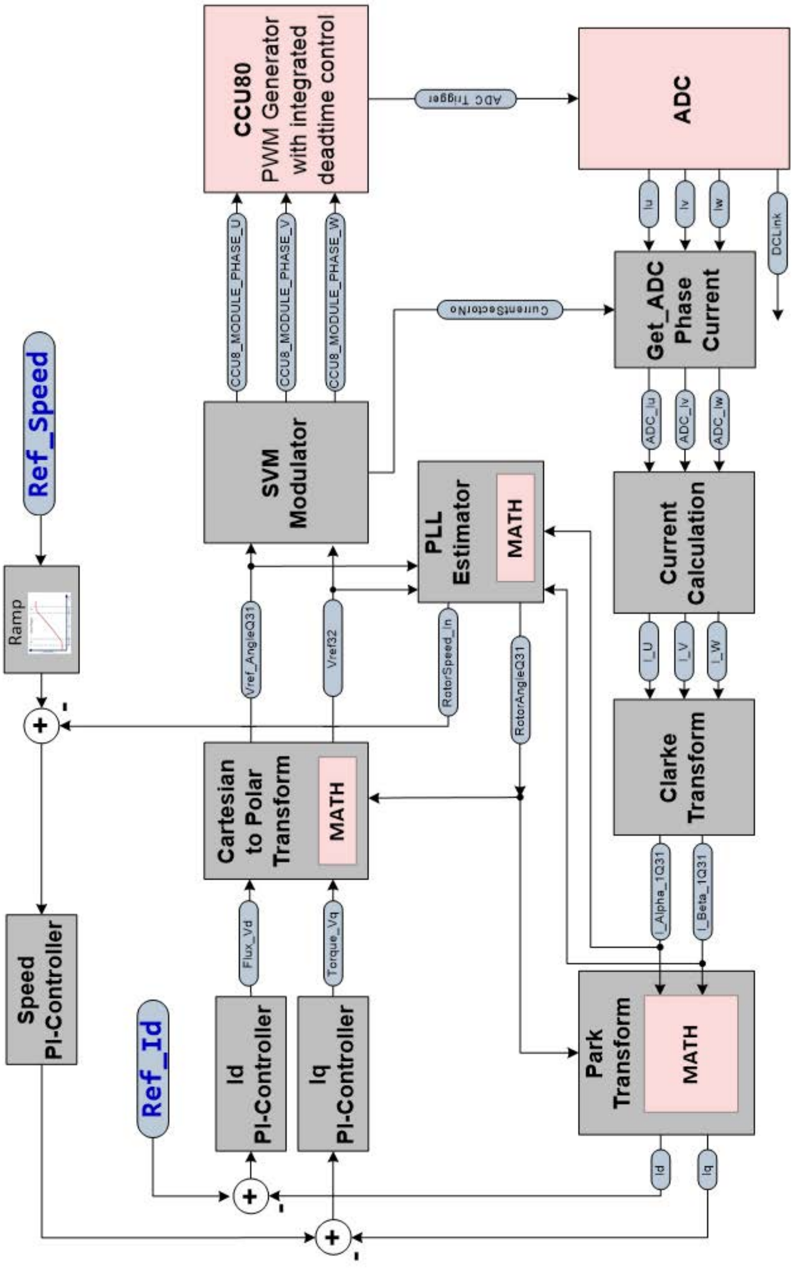


Figure 6.4: Speed FOC Scheme

Torque Control Direct FOC Startup

In this scheme the controlled parameter is represented by I_q, which is proportional to the torque. Since it does not rely on the rotor speed, in only uses currents for closing the feedback loops. Furthermore, as explained in chapter 4, the I_d current is controlled with reference to 0. In figure 6.5 it is shown the resulting scheme.

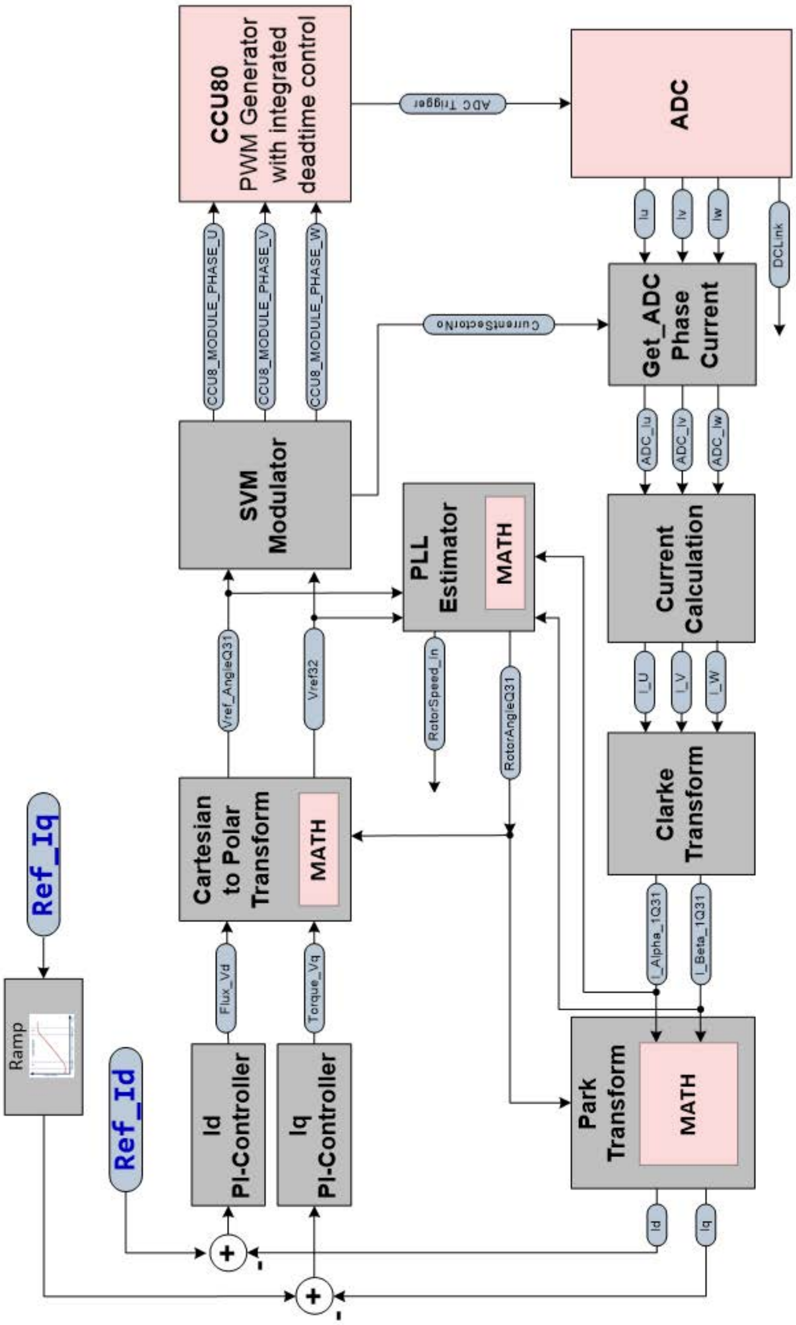


Figure 6.5: Torque FOC Scheme

Vq Control Direct FOC Startup

In this scheme V_q, which is the voltage component perpendicular to the rotor, is imposed via a ramp. This kind of control is used when a fast response is required and controlling the speed is not a requirement. In this configuration, I_d is control in closed-loop and is equal to 0, as shown in figure 6.6.

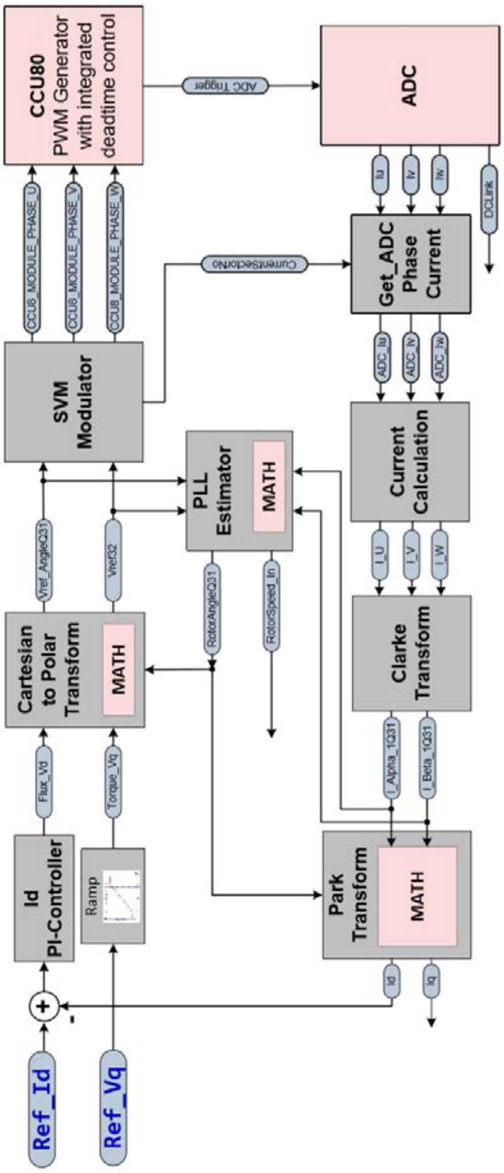


Figure 6.6: Vq scheme

Speed Control Transition FOC Startup

This configuration is obtained as a a combination of two configuration: the speed FOC and the V/Hz control. Thanks to them, the controller start the motor with a V/Hz regulation and switches to a FOC closed loop when the rotor speed is sufficiently high.

6.2 Software Structure

The PMSM FOC motor control software is based on a well-defined layered approach, shown in figure 6.7.
The layered architecture is designed to separate the modules into groups, thus allowing different modules to be easily replaced without affecting the performance of other modules and the structure of the complete system.

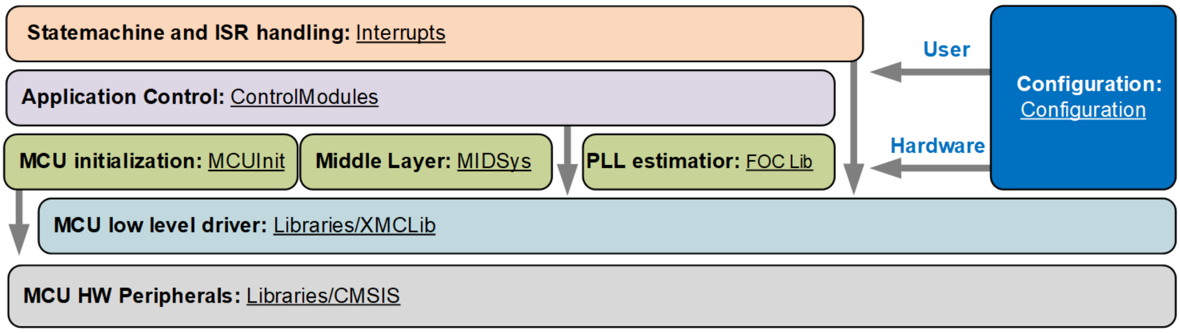

Statemachine and ISR handling: Interrupts
Application Control: ControlModules
MCU initialization: MCUInit
Middle Layer: MIDSys
PLL estimatior: FOC Lib
MCU low level driver: Libraries/XMCLib
MCU HW Peripherals: Libraries/CMSIS
User
Hardware
Configuration: Configuration

Figure 6.7: Software Layers

State machine and ISR handling: Interrupts

This layer consists of a CCU8 trap interrupt handling function, a cycle machine function, and a VADC interrupt for under/overvoltage control.

Application Control: Control Modules

This layer consists of FOC software control modules. This includes the Clarke Transform, Park Transform, Cartesian to Polar, current reconstruction, PI controller, Open Loop and Ramping for example.
All the routines mentioned are called from the CCU80 period match Interrupt Service Routine.

Configuration: Configuration

In this layer, all the user configuration options needed by the FOC firmware are defined, including controller card, inverter card and motor specifics. Furthermore, the user configuration options affects the general behavior of the software like PWM generation, scheme control and safety.

PLL estimator: FOCLib

The PLL estimator is an Infineon patented IP, and is provided as a compiled library file. No further informations can be shared about this module.

Middle Layer: MIDSys

This layer provides routines for PWM generation and ADC measurements to the FOC control module layer. The main purpose of this layer is to give flexibility to add or remove a sensor feedback module into the FOC software. This layer provides also a mathematical library for trigonometric operations.

MCU Initialization: MCUInit

This layer controls the initialization of all MCU peripherals, as ADC, CCU8 and CCU4. It contains XMCLib data structure initialization and peripherals initialization functions. This layer closely interacts with XMCLib and the MIDSys layer to configure the microcontroller.

MCU low level driver: Libraries/XMCLib and MCU hardware Peripherals: Libraries/CMSIS

This layer is the hardware abstraction layer to the MCU peripherals. Thanks to it, it is not needed to work on registers but it provides APIs to simplify operating with the microcontroller.

6.3 Low-level and Middle-level layers

The low level layer is composed by MCUInit and MIDSys layers. It consists in drivers and basic routine, like current sensing, SVM calculatios and math functions.

6.3.1 Driver and Hardware setting

The field oriented control moduls uses 4 CCU8 slices, one CCU4 slice, 3 ADC and a watchdog to work properly. Of the four CCU8 slices, three are used to generate the PWM signals that drive the inverter, while the fourth is used to generate the trigger for the ADC.

CCU8 Configuration

As already mentioned, three of the four slices are used for generating the PWM signal. Furthermore, the hardware also generates the negated output since it is required by the three-phase inverter. Each PWM and its relative negated signal drive the high-side and the low-side of the inverter leg. Depending on the functioning of the driver, that could be active high or active low, PWMs are generated in positive or negative logic. A dead time is inserted between the PWM and its relative negated signal to prevent a short-circuit and heat generation on the inverter leg. The Phase U slice gives the timing to the machine cycle throwing an interrupt every period match event.
The fourth slice is used to trigger the ADC with a compare match interrupt, synchronized with the PWM signals.
An overcurrent signal generated by the inverter is connected to the trap of each CCU8 to prevent damages. Furthermore, Phase U CCU8 slice throws an interrupt to the microcontroller when trap signal rises. The interrupt handler stops the FOC operations and it handles the overcurrent by stopping the generation of PWM and disabling the inverter. All 4 slices generates PWM signals at the same frequency, 20000Hz. This particular frequency has been chosen because it is higher than the sound threshold, thus it can not generate audible noise.

CCU4 Configuration

Only one slice of CCU4 is used to generate a period match interrupt for running a second and slower cycle used for less important operations. The cycle runs at 1000Hz.

VADC Configuration

In the VADC sampling requests are dived into two categories:

- In queue source, the measuring of the phase currents through the shunt resistances in a synchronous way. The trigger from the fourth CCU8 slice is used here to generate the sampling request.

- In background source, the measuring of the DC-link voltage in order to detect over/under voltage situations.

6.3.2 Space vector modulation module

In this module it is implemented the Space Vector Modulation (SVM), used in all the control schemes. Starting from the polar coordinates of the voltage, the module calculates the duty cycle for each of the PWM signals. In figure 6.8 it is represented the module.

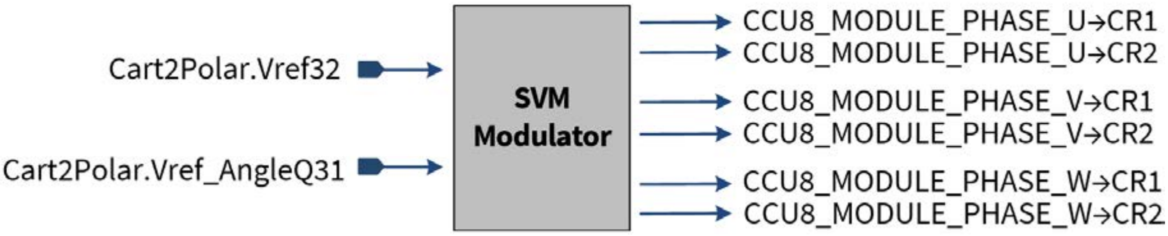


Figure 6.8: SVM Module

The developed PMSM FOC motor control software supports different modes of generating PWM signals:

- 7-segment SVM
- 5-segment SVM
- SVM with Pseudo Zero Vector
- 4-segment SVM
- Over-modulation SVM

7-segment SVM

The 7-segment modulation consists of dividing the SVM period, T_S, in 7 time slots and assigning to each one an active or passive vector. With reference to fig. 6.9, the active vectors are $\vec{V_1}$,$\vec{V_2}$,$\vec{V_3}$,$\vec{V_4}$,$\vec{V_5}$ and $\vec{V_6}$, thus the voltage vector $\vec{V_{ref}}$ can be generated by the two active vectors $\vec{V_1}$ and $\vec{V_2}$ while the passive vectors are represented by $\vec{V_0}$ and $\vec{V_7}$. The time period of each active vector, T_1 and T_2, can be seen geometrically as proportional to the projections of the vector $\vec{V_{ref}}$ on the active vectors, as reported in figure 6.9. The remaining time, T_0, is used to generate zero vectors.

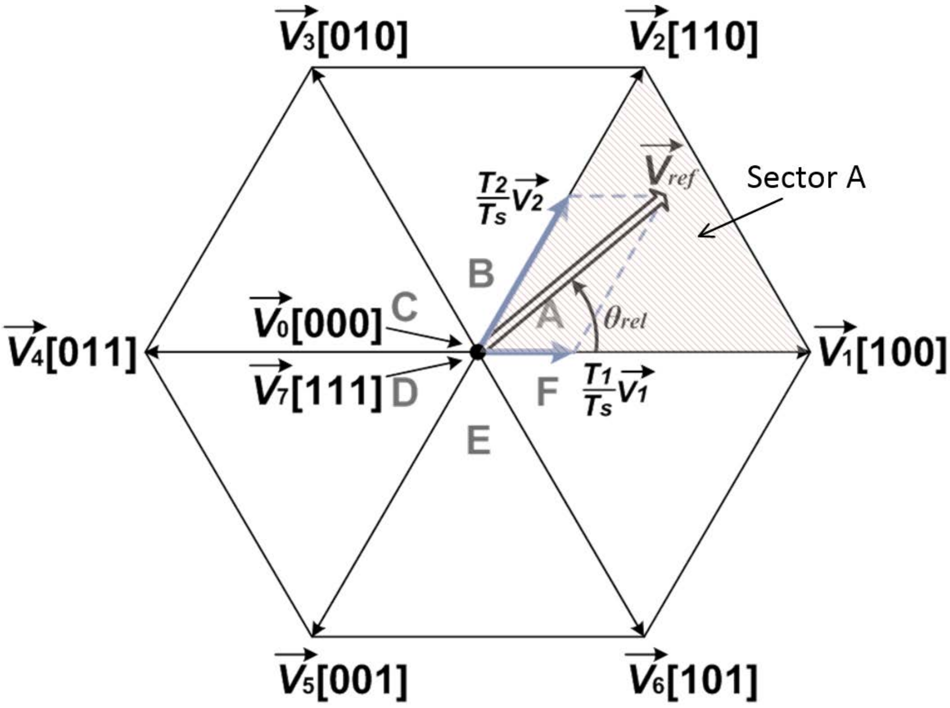


Figure 6.9: 7-segment time calculations

The following equations ha been used to calculate the times analytically:

$$\vec{V}_{ref} = \frac{T_0}{T_S}\vec{V}_0 + \frac{T_1}{T_S}\vec{V}_1 + \frac{T_2}{T_S}\vec{V}_2 \tag{6.1a}$$

$$T_S = T_0 + T_1 + T_2 \tag{6.1b}$$

$$T_1 = \frac{\sqrt{3}|V_{ref}|T_S}{V_{DC}} sin\left(\frac{\pi}{3} - \theta_{rel}\right) \tag{6.1c}$$

$$T_2 = \frac{\sqrt{3}|V_{ref}|T_S}{V_{DC}} sin\left(\theta_{rel}\right) \tag{6.1d}$$

$$T_0 = T_S - T_1 - T_2 \tag{6.1e}$$

Where θ_{rel} is the voltage vector angle and V_{DC} is the supply voltage.

In figure 6.10 it is shown for each slot the associated vector and its period. The first slot is assigned to [111] zero vector and it has a period of $T_0/4$, while the next two slots are assigned to the second and the first active vectors, with periods equal to the half of, respectively, T_2 and T_1. The central slot is destined to [000] zero vector for a time equal to $T_0/2$, then the vectors of the first three slot are repeated in opposite symmetry. Therefore, in the last slots there are the first active vector for $T_1/2$, the second active vector for $T_2/2$ and the [111] zero vector for $T_0/4$. In figure it is also shown an example of Phase U counters values CR1 and CR2, calculated by the SVM modulator, and their correlation with the PWM signal.

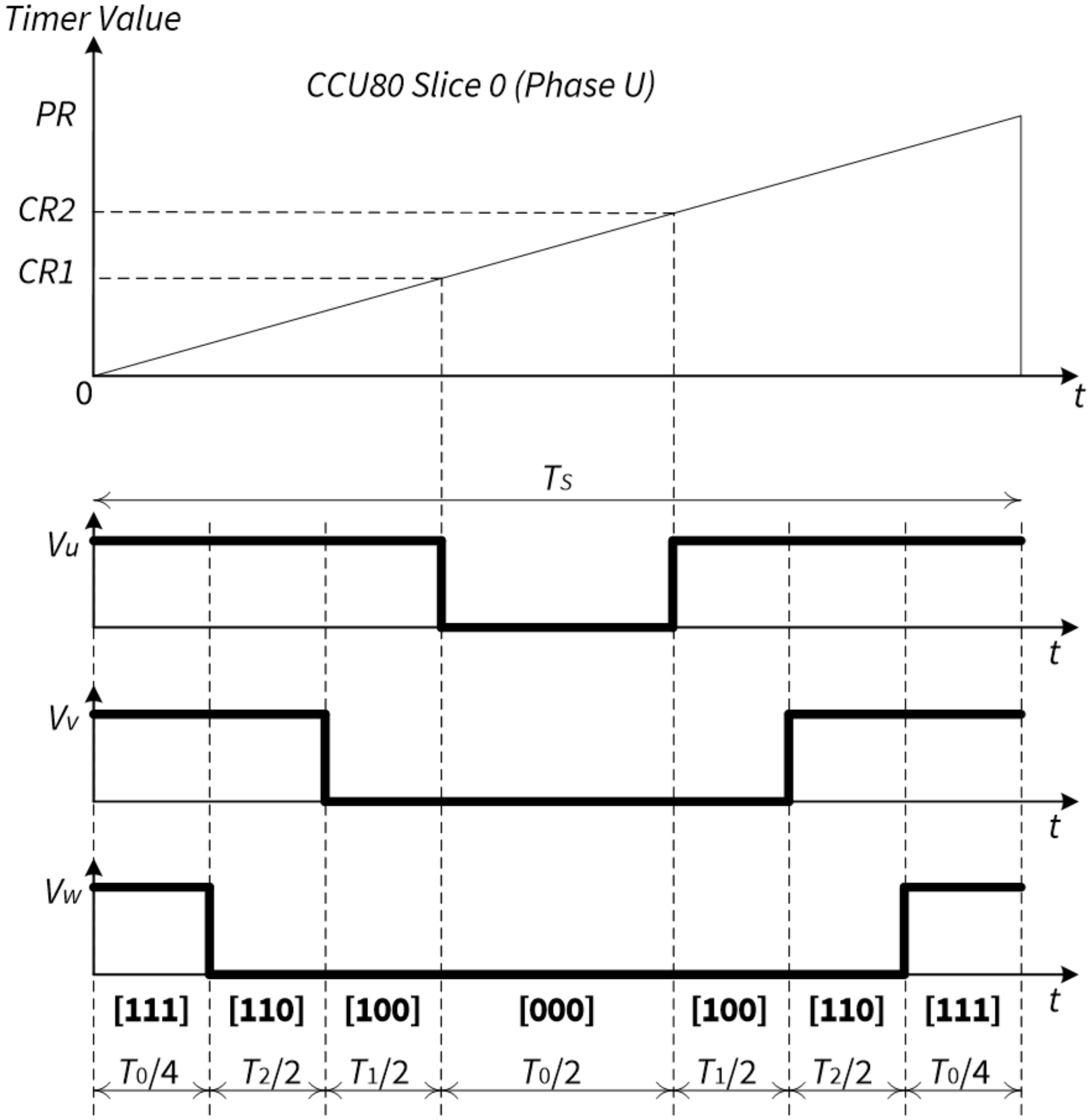


Figure 6.10: 7-segment PWM

5-segment SVM

The 5-segment SVM uses the same equations as in 7-segment SVM (6.1) to calculate T0, T1, and T2 timing, but in this modulation the first and last slots are removed, thus the zero vector is assigned to the central slot for a time equal to T_0, as shown in figure 6.11.

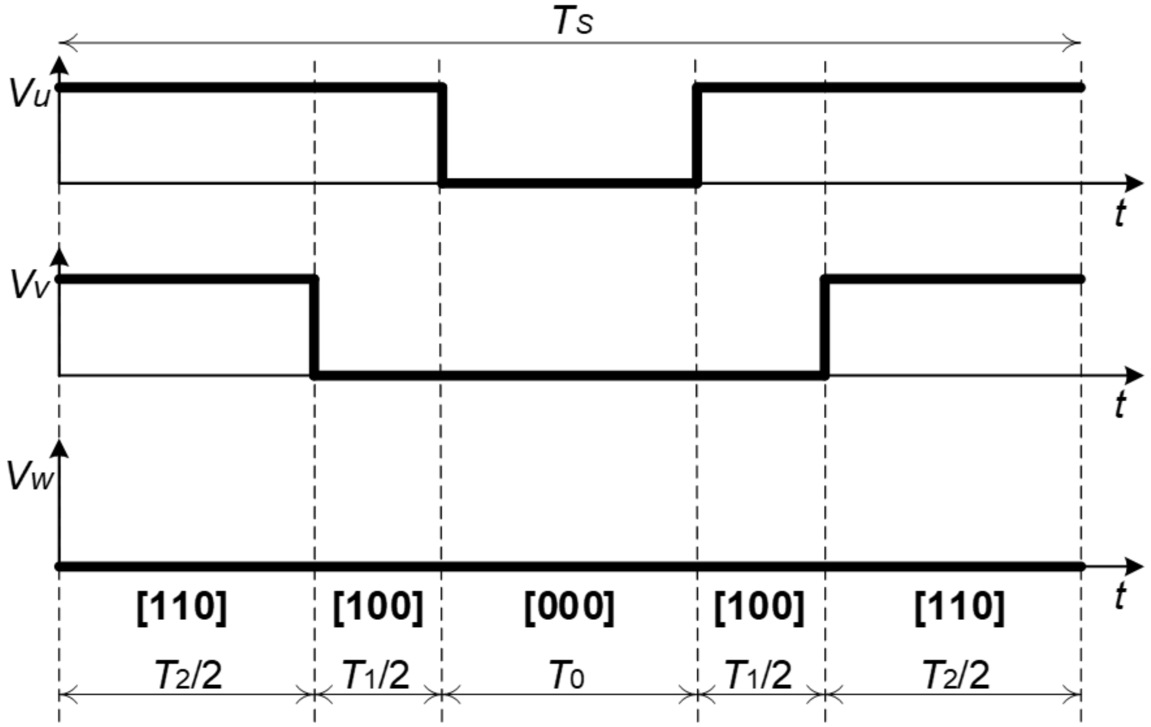


Figure 6.11: 5-segment PWM

Pseudo Zero Vector (PZV)

In this modulation, the zero vectors can be generated combining two opposites active vectors, where one has the same period of the other. Thanks to this, their sum does not affect the resulting voltage vector since they cancel each other.
For example, with reference to figure 6.12, the active vector that generate the desired voltage are [110] and [100], but their times are longer than necessary because they are also used to generate pseudo zero vectors. Therefore, they are compensated by their opposite vectors [001] and [011] activated for a period equal to this extra time.

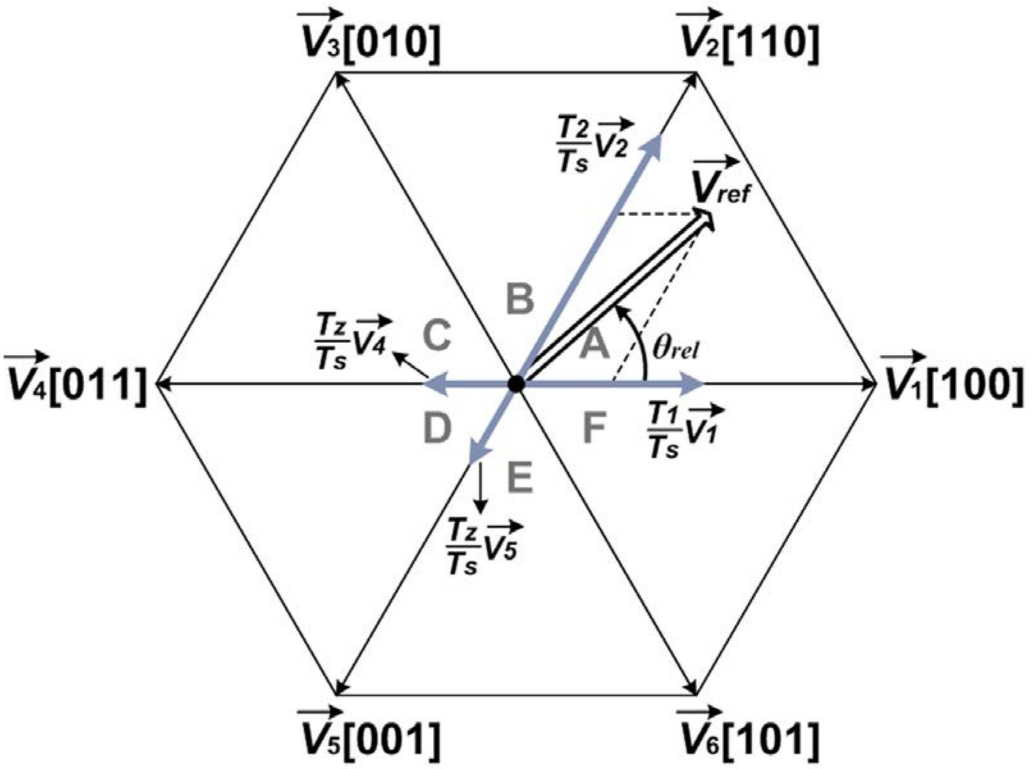


Figure 6.12: PZV time calculations

This method is useful when each state must have a minimum time (T_{min}). The formulas used to calculate the timings are:

$$T_1 = \frac{\sqrt{3}|V_{ref}|T_S}{V_D C} sin\left(\frac{\pi}{3} - \theta_{rel}\right) + T_{min} \tag{6.2a}$$

$$T_2 = \frac{\sqrt{3}|V_{ref}|T_S}{V_D C} sin\left(\theta_{rel}\right) + T_{min} \tag{6.2b}$$

$$T_0 = T_S - T_1 - T_2 \tag{6.2c}$$

Where T_{min} is the minimum active time for each vector.
With this conditions, the generated internal area of the hexagon can be smaller. The higher T_{min} is, the smaller the area that can be generated is, as shown in figure 6.13.
In figure 6.14 it is shown PWM signals example of this type of modulation.

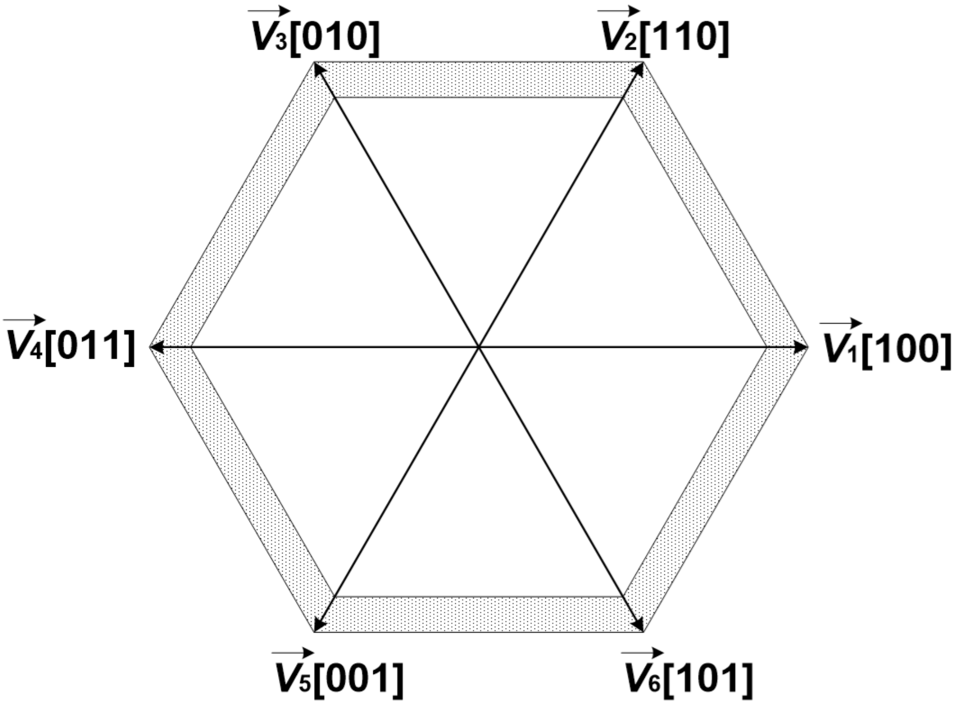


Figure 6.13: PZV hexagon area restriction

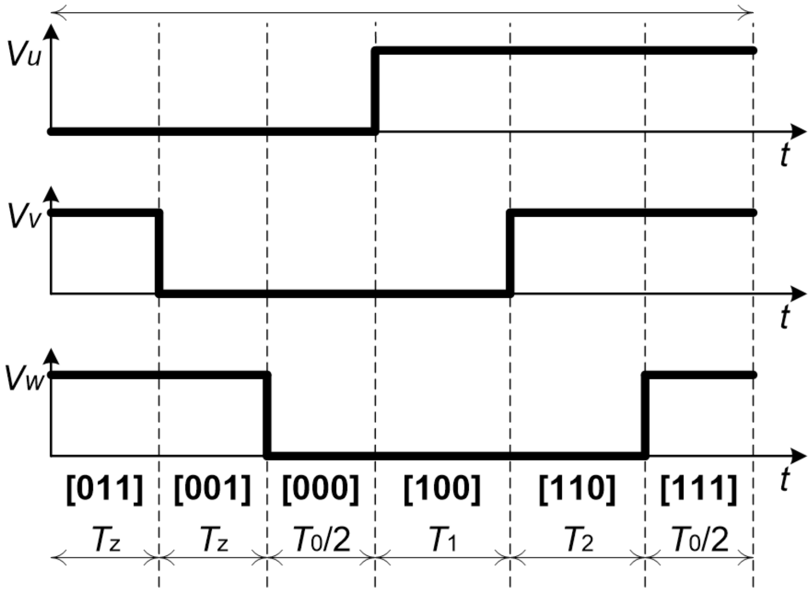


Figure 6.14: PZV pwm

4-segment SVM

The 4-segment SVM is a further simplification of the 5-segment SVM, where the SVM period is splitted in 4 slots. The combination of vectors consists in [000] in the first slot for a time equal to $T_0/2$, then the first active vector followed by the second active vector respectively for T_1 and T_2 and, finally, the zero vector [111] is set for $T_0/2$, as shown in figure 6.15.

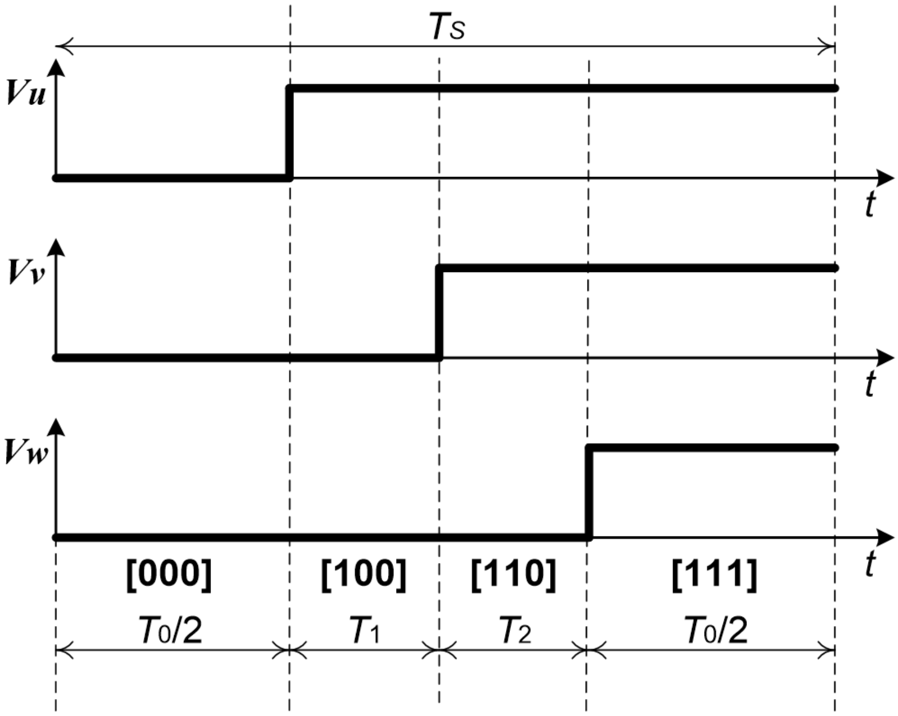


Figure 6.15: 4-segment PWM

Overmodulation SVM

In SVM, for sinusoidal commutation, V_{ref} has to be smaller than 86% of its maximum value, but using overmodulation it is possible to generate non-sinusoidal commutation using V_{ref} with its maximum amplitude.
Figure 6.16 shows the area, shaded in red, where the over-modulation technique is used.

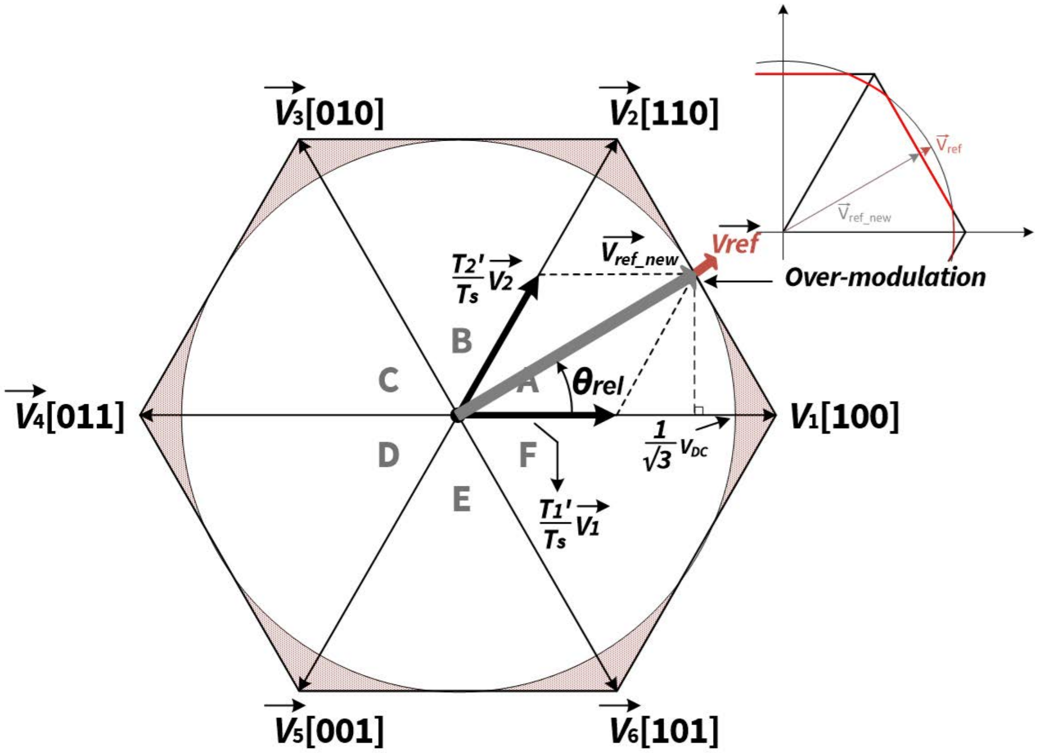


Figure 6.16: Overmodulation Area

In the developed PMSM FOC firmware, the over-modulation is implemented by using a rotating voltage vector, V_{ref}, with the maximum voltage module available and reducing the amplitude to V_{ref_new} when it exits the hexagon area. Analytically this happens when the sum of T_1 and T_2 is higher than PWM period T_S, and it is resolved by reducing the T_1 and T_2 timing proportionally. Therefore, T_1 and T_2 are re-calculated as:

$$T_{1\_new} = T_1 \times \frac{T_S}{(T_1 + T_2)} \tag{6.3a}$$

$$T_{2\_new} = T_S - T_{1\_new} \tag{6.3b}$$

In over-modulation, the time for zero vector is reduced to zero, therefore when the motor is running at high-speed, over-modulation is used to maximize the power delivered by the motor. In the other hand, the over-modulation output is not sinusoidal, and it contains high-order harmonics that can cause acoustic noise.
In figure 6.17 it is shown the resulting PWM signals.

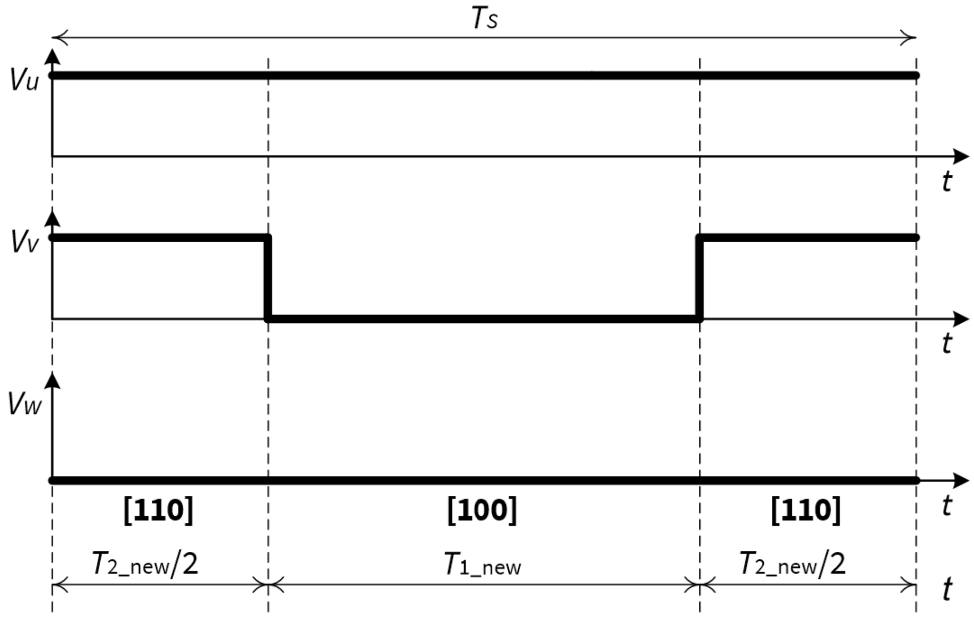


Figure 6.17: Overmodulation PWM

6.3.3 Current sensing and trig synchronization

The three-phase current is reconstructed by the FOC firmware by sampling the voltage drops that occur on the three resistances placed in each of the half-bridge legs. The voltage drops are proportional to the phase currents and occur only when they flow through the shunt resistance, thus when low-side transistors are conducting. In the developed PMSM FOC firmware, it has been chosen to implement either the 5-segments SVM and the 7-segments SVM because the PWM signals in these two modulations have the central part equal to 0, thus the low-sides of the legs are active at the same time and currents flow can be sampled synchronously.
The ADC trigger, provided by CCU8 trig slice, should then start at half of T_S period which corresponds to a duty cycle of the slice equal to 50%, as shown in figure 6.18, where it can also be seen an example of generated phase currents.

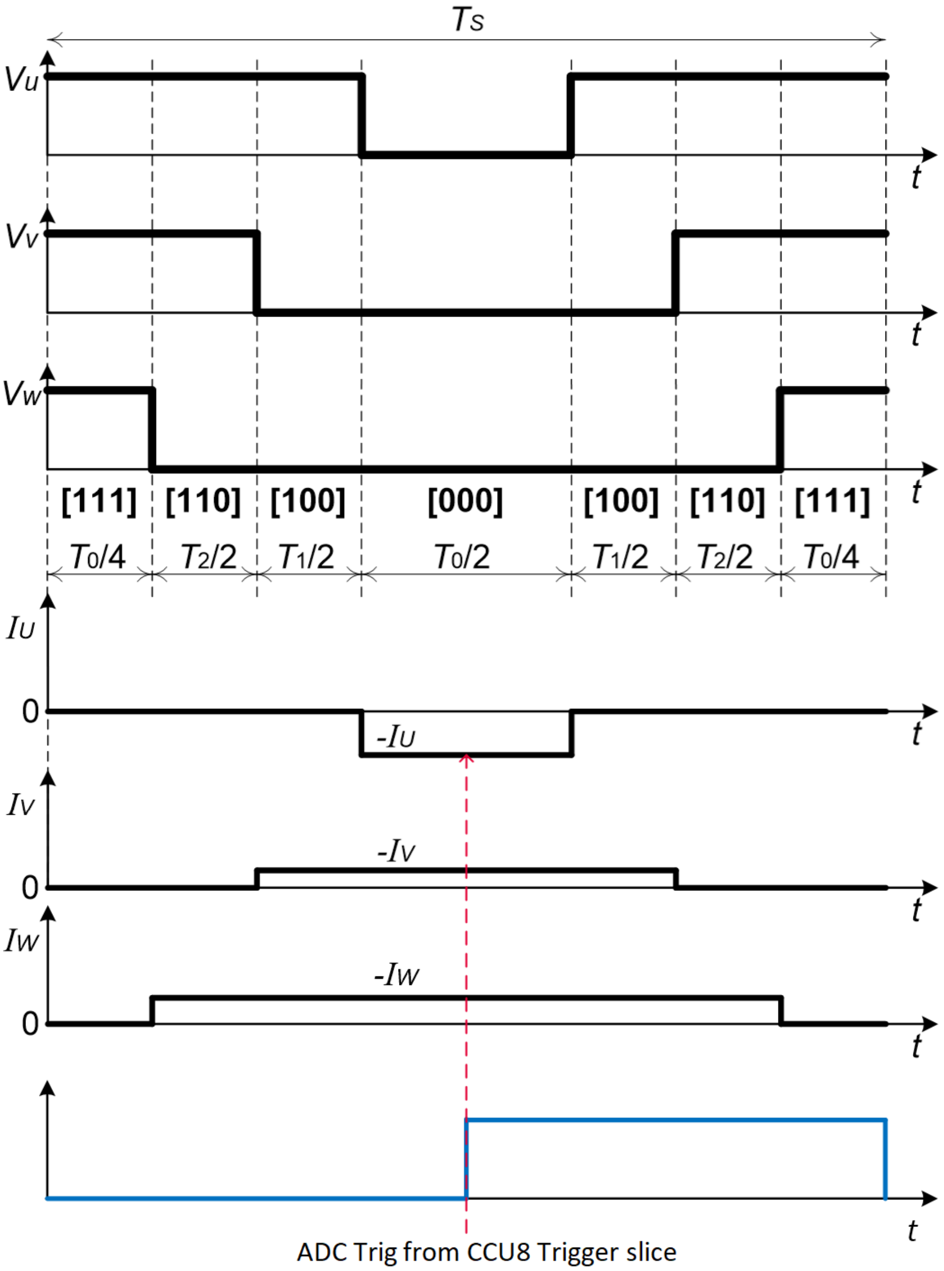


Figure 6.18: ADC Trigger

The other SVM modulations can be used in open-loop control, where is not necessary to read or estimate the rotor position.

6.3.4 Math library

The mathematical functions needed by the Field Oriented Control algorithm to work properly are implemented in the math library. The library uses the FPU unit of the XMC4400 microcontroller CPU implementing sine and cosine functions and the previously mentioned cartesian to polar and Park transform.

Sine and Cosine

Two versions of the sine and cosine functions have been considered in this work. The first one, uses the functions implemented in the CMSIS library, which returns a Q31 value for a given angle. A second version of the two functions has been implemented to further reduce the computation time, that are based on a look-up table and exploit the FPU unit by only using float data types.

Park Transform

The Park transform function implements the Park Transform presented in the section 4.1.2 using the sine and cosine functions here explained.

Cartesian to polar Transform

The cartesian to polar transform calculates the module and the phase of a vector starting from its cartesian components.
To calculate the magnitude, it uses the square root of the components squares sum, while to calculate the phase the atan function is used. Since it is impossible to map into a lookup table the infinite and non-periodic domain of the atan function, the library uses a mix of LUT and math approximation to calculate it. For the atan LUT, 300 variable step samples are used in the domain from 0 to a fixed value (set to 5.02 in the library). In figure 6.19 it is shown the samples distribution of the implemented atan function.

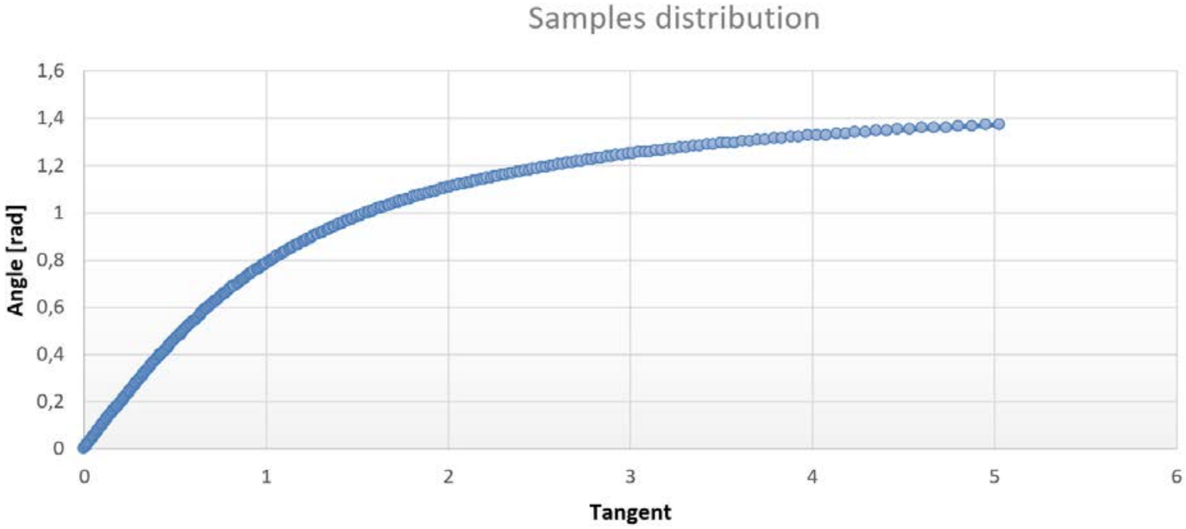


Figure 6.19: Arctangent samples distribution

A variable step size has been used to map the domain with more points where the function is more curved. The mathematical map to convert tangent value to the sample vector index is in relation with the derivative of arctangent function, properly scaled and adapted:

$$y = \frac{\left(\frac{x}{X_{scaling}} + X_{offset}\right)^2}{1 + \left(\frac{x}{X_{scaling}} + X_{offset}\right)^2} \cdot Y_{scaling} + Y_{offset}$$

Where x is the tangent value, y is the index vector and $X_{scaling}$, X_{offset}, $Y_{scaling}$ and Y_{offset} are scaling and offset factors for the domain and the codomain.
Using this map it is possible to obtain a floating point index for a given angle, and using linear interpolation, it can be calculated the relative angle value. Because of the concavity of the atan function, the linear interpolation would always return an underestimated value, thus a constant is added to all samples to compensate this phenomenon (AT_CC_ERROR).
For tangent values larger than the last LUT element, the library uses the following math approximation:

$$y = -\frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} + \frac{\pi}{2}$$

Where x is the tangent value and y is the resulting angle in radiants.
The precision of the atan function is about 2E-6 radiants (0.00012 degrees, 20 bit in q31) and the library also support an error correction that bring the precision to 5E-7 radiants (0.00003 degrees, 22 bit in q31).

6.4 High Level Layer

The FOC modules are implemented in the high level layer of the firmware. The machine cycle is called through a period match interrupt generated by the phase U CCU8 slice while another slower interrupt generated by the CCU4 calls a routine that executes less important tasks, like reading input.

6.4.1 PLL Estimator (Observer)

A Infineon patent called PLL estimator is used to implement the observer. It is able to estimate position and speed starting from the polar coordinates of the three-phase vector that drives the PWM and the $\alpha\beta$ coordinates of the read currents.

6.4.2 Interrupts

Primary Loop - Machine cycle

The primary loop is executed at the same frequency of the PWM signals and it is where the FOC calculations are performed. The interrupt that calls this loop is given by the period match event of the U phase PWM slice.
When the motor is running, the measurements of the last currents samples are read from the ADC registers and the Clarke and Park transforms are performed obtaining the currents in the dq plane. The results are then processed by the PI and the SVM is executed to calculate the PWM duty cycle values.
In addition, the PLL observer operations are performed.
If the control scheme is set to speed-control, the second feedback loop is performed, therefore the estimated speed and the target speed are compared and the relative PI controller sets the new I_q reference.

Secondary Loop

This loop runs at a lower frequency with respect to the primary loop. During it execution, the input targets are read (for example the target speed) and it is checked if there is a request to start or stop the motor. The values of the V_{DC} link are then read and the watchdog is fed. The trig is given by the period match event of a CCU4 slice.

Trap Interrupt

This interrupt is generated by the Phase U CCU8 slice, based on the TRAP signal received from the inverter. When an overcurrent is detected, inverter trap pin is activated and the CCU8, which is connected to it, stops generating signals and throws an interrupt. The interrupt handler disables the inverter to protect it from damage.

Under/Over voltage Interrupt

This interrupt is generated by the ADC connected to the V_{DC} link. When the ADC reads values of the V_{DC} link outside the prefixed boundaries, it throws an interrupt and the interrupt handler disables the inverter to protect it from damage.

6.4.3 State Machine

The developed software can be represented as a finite-state machine, which its state depends on the configuration and the inputs given.
In figure 6.20, it is reported the finite-state machine model of the developed firmware. For different control schemes, the flow of the state machine is different. The possible states are:

- MOTOR_IDLE: This is the first state entered after power-on or on software reset. In this state the inverter is disabled and firmware reads the bias voltage of the ADC pins that are connected to the motor phase currents, then the state machine and timer are started. When the motor start command is received, the state is changed to EN_INVERTER_BOOTSTRAP.

- EN_INVERTER_BOOTSTRAP: In this state the inverter is enabled and FOC firmware reads the bias voltage of the ADC pins that are connected to the motor phase currents.
  After the bootstrap time, the state can change either to PRE_POSITIONING or to VF_OPENLOOP_RAMPUP, depending on the selected scheme control.

- PRE_POSITIONING: This state is only for Direct FOC Startup control schemes. In this state the rotor is aligned to a known position to get the maximum starting torque while the amplitude voltage is gradually increased to a defined voltage value, for a specific time. At the end, the state is changed to FOC_CLOSED_LOOP.

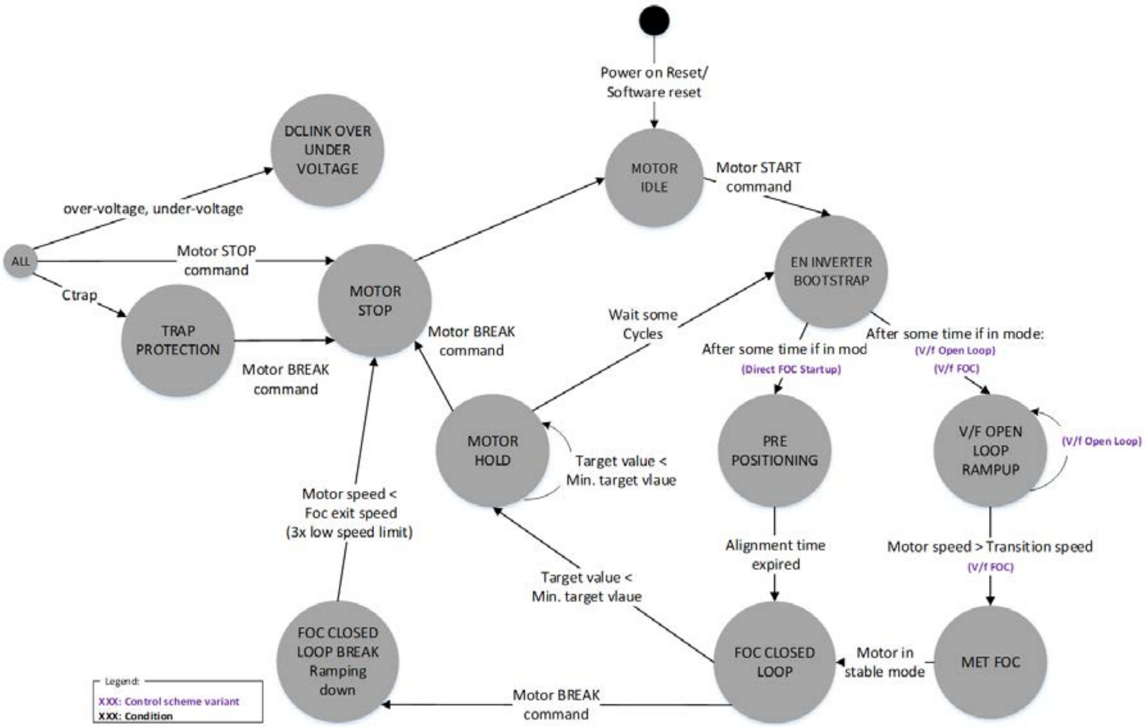


Figure 6.20: State Machine[9]

- VF_OPENLOOP_RAMPUP: In this state the motor starts in V/F open loop control mode. When the motor speed reaches the startup threshold speed defined by the user, the state goes into MET_FOC.

- MET_FOC: This state enables a smooth transition from open loop to closed loop with maximum energy efficiency, after which the state is transitioned to FOC_CLOSED_LOOP. To exit from this state it must be generated a stop command or the speed must be set below the minimum target value.

- FOC_CLOSED_LOOP: In this state the motor is running in FOC mode and the FOC functions are executed.

- FOC_CLOSED_LOOP_BREAK: This state needs when a stop command is received. The FOC is still working but the speed target value is ramping down in an S-curve and after crossing the FOC_EXIT_SPEED the state is changed to MOTOR_STOP.

- MOTOR_HOLD: This state is entered when the motor speed is below 10% of its maximum speed. The motor phases are connected to ground and no power is delivered. A Motor break command in this state leads directly to a MOTOR_STOP.

- MOTOR_STOP: This state is entered from all states with the stop command, the motor output is set to tristate and the inverter is disabled causing an uncontrolled freewheeling of the motor. The state exits to the idle state when the the motor is stopped.
- TRAP_PROTECTION: This state is entered if the CCU8 trap is triggered and to exit from this state a stop command is needed.
- DCLINK_UNDER_VOLTAGE: This state is entered when the DC link voltage is below the limits set by the user. The gate driver is disabled and the motor is set to free running and only the motor stop or motor brake command can exit this state.
- DCLINK_OVER_VOLTAGE: This state is entered when the DC link voltage is above the limits set by the user. The gate driver is disabled, the motor is in free running and only the motor stop or motor brake command can exit this state.

6.4.4 PI control

One of the main modules of the developed firmware is the proportional–integral controller, because it processes the feedback loop and drive the controlling variables accordingly. The implemented regulator also features an antiwindup mechanism that helps the integral to not overload when saturation of the outputs occurs. With reference to figure 6.21, to calculate the initial values of the PI gains of the torque and flux currents control, it is necessary to know the motor's model parameters, considering that, for simplicity, in a PMSM motor the torque inductance and the flux inductance are considered equal.

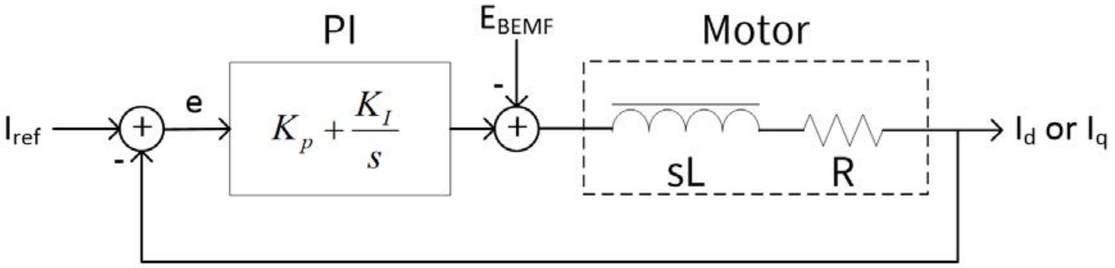


Figure 6.21: PI loop

The calculation of the PI gains is made by using the pole-zero cancellation technique, thus the controller zero cancels the motor pole by having:

$$\frac{K_P}{K_I} = \frac{L}{R}$$

Where K_P is the proportional gain, K_I is the integral gain, L is the motor inductance and R the motor resistance.
Using these parameters in the PI transfer function, the resulting closed loop transfer function is a first order low pass filter with time constant T_c. At constant motor speed the Back-EMF of the motor is near constant, therefore it is negligible in the frequency domain. The figure 6.22 shows the simplified diagram after pole-zero cancellation.

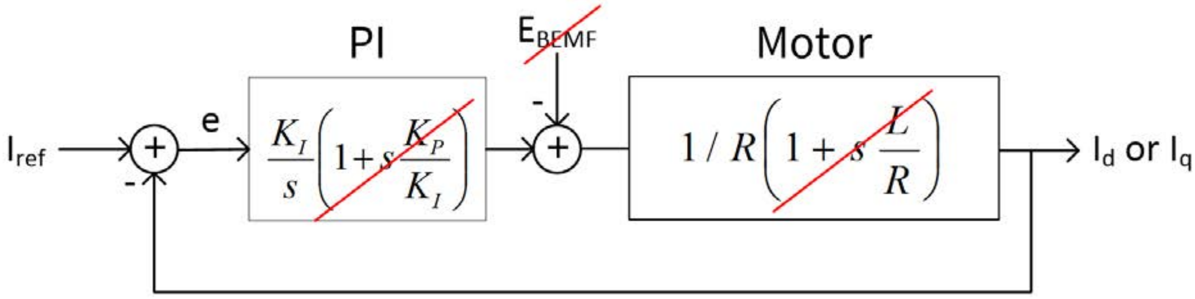

PI
E_BEMF
Motor
I_ref
e
+
-
$\frac{K_I}{s}\left(1+s\frac{K_P}{K_I}\right)$
$1/R\left(1+s\frac{L}{R}\right)$
I_d or I_q

Figure 6.22: PI cancellation

As $K_P/L = K_I/R = \omega_c$, the PI controller gains can be calculated as:

$$\text{Proportional Gain } K_P = \omega_c L \tag{6.4a}$$
$$\text{Integral Gain } K_I = \omega_c R \tag{6.4b}$$

Where ω_c is the cutoff frequency of the first order low pass filter.
In the digital controller implementation, the integral part is a digital accumulator, therefore the K_I gain has to include a scaling factor for the sampling time T_S, which is the PWM frequency. Therefore the formulas 6.22 can be written as:

$$\text{Proportional Gain } K_P = \omega_c LA \tag{6.5a}$$
$$\text{Integral Gain } K_I = \omega_c RT_S = (RT_S K_P)/L \tag{6.5b}$$

Where A is the XMC hardware optimize scaling factor. A general rule is to set the cutoff frequency to three times of the maximum electrical motor speed to obtain a good tradeoff between dynamic response and sensitivity to the measurement noise.

7 Performance Test

Once the implementation was completed, it was proceeded to carry out measurements on the behavior of the FOC with the aim of verifying if the execution speed and accuracy requirements have been respected.

7.1 Functionality test

The hardware setup used during the tests was composed by a microcontroller board Infineon KIT_XMC4400_DC_V1 and an inverter Infineon KIT_MOTOR_DC_250W_24V, while the motor was a Nanotec DB42S03. A software oscilloscope, the Micrium uC-Probe XMC, was used to read the digital value of the physical quantities from the microcontroller memory.
The object of this test is to verify the precision of the FOC control carrying out measurements of I_U, I_V and I_W phase currents, I_d and I_q currents and speed. The results are shown in figures 7.1 and 7.2.

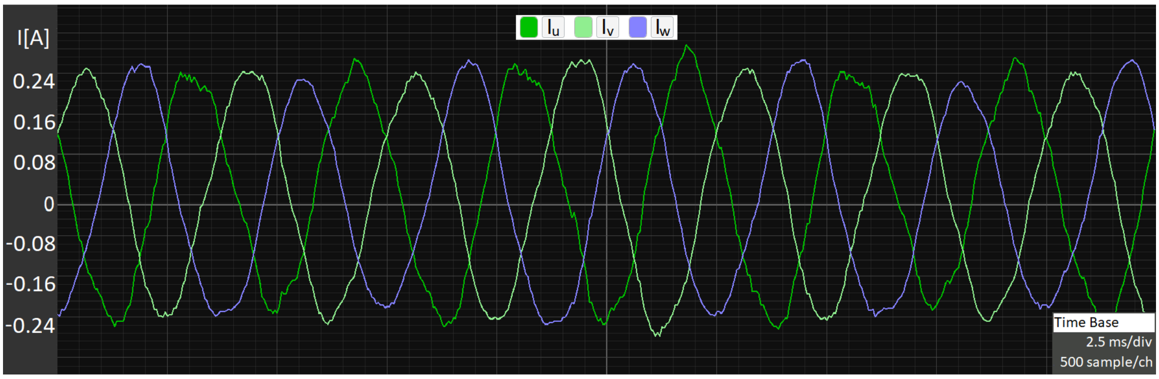


Figure 7.1: Measured I_U, I_V and I_W phase currents

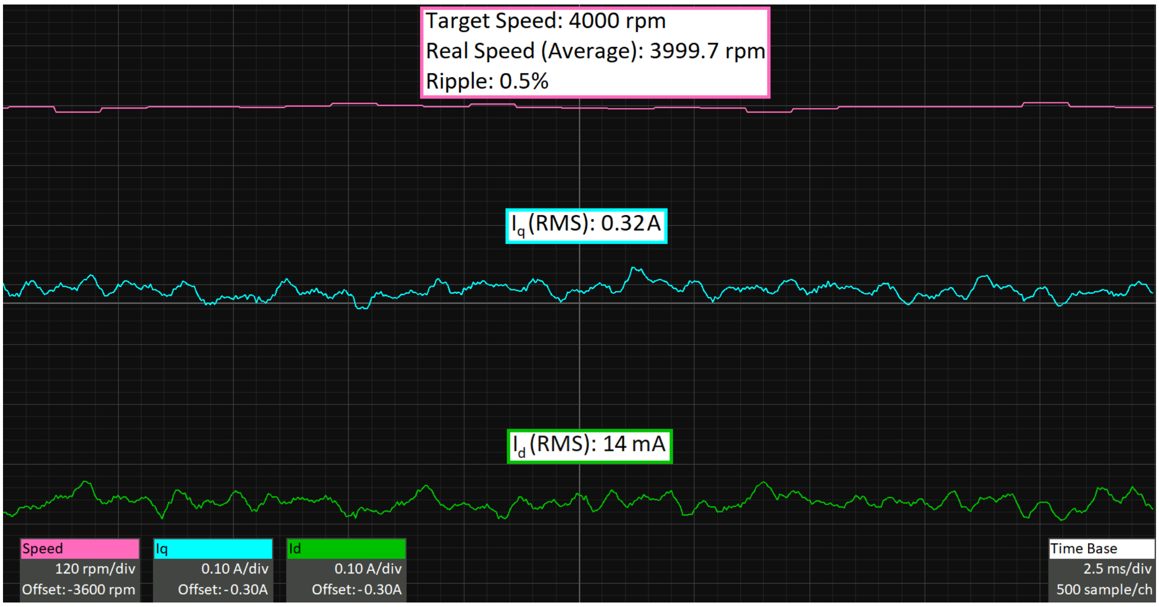


Figure 7.2: Measured I_d and I_q currents and speed

As it can been seen, as expected, the I_u, I_v and I_w current signals represent sinusoidal signals shifted of 120°, while the speed was kept stable by the FOC with a ripple of 0.5%. The I_d was almost zero while the I_q was appropriate for the applied load.
From the data obtained, it was concluded that the developed Field Oriented Control firmware works correctly.

7.2 Speed test

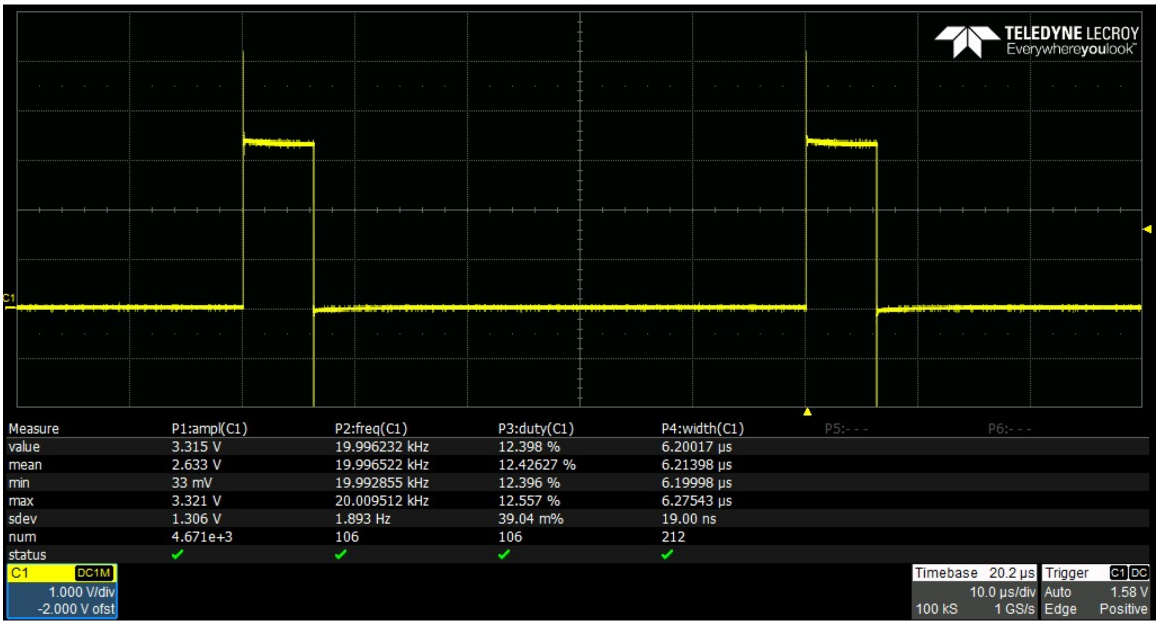
In this test, it was proceeded to verify the execution speed of the Field Oriented Control firmware by carrying out measurements on the execution speed.
The measurement is made by raising up and lowering down a GPIO pin in the primary interrupt, therefore connecting an oscilloscope to it and measuring the duty cycle.
The duty cycle represents the percentage of time spent during the machine cycle to carry out defined instructions.
The oscilloscope used in these tests is a Teledyne Lecroy WaveSurfer 3054z with 500MHz bandwidth and the measurements made are:

- The Park transform execution time
- Cartesian to polar transform execution time

- The total execution time of the control

7.2.1 Park Transform Execution Time

In the first implementation of the Park transform, it was used the CMSIS library to calculate the sine and cosine functions, which uses 32-bit and 64-bit integer data types.



Measure	P1:ampl(C1)	P2:freq(C1)	P3:duty(C1)	P4:width(C1)	P5:- - -	P6:- - -
value	3.315 V	19.996232 kHz	12.398 %	6.20017 µs		
mean	2.633 V	19.996522 kHz	12.42627 %	6.21398 µs		
min	33 mV	19.992855 kHz	12.396 %	6.19998 µs		
max	3.321 V	20.009512 kHz	12.557 %	6.27543 µs		
sdev	1.306 V	1.893 Hz	39.04 m%	19.00 ns		
num	4.671e+3	106	106	212		
status	✓	✓	✓	✓		

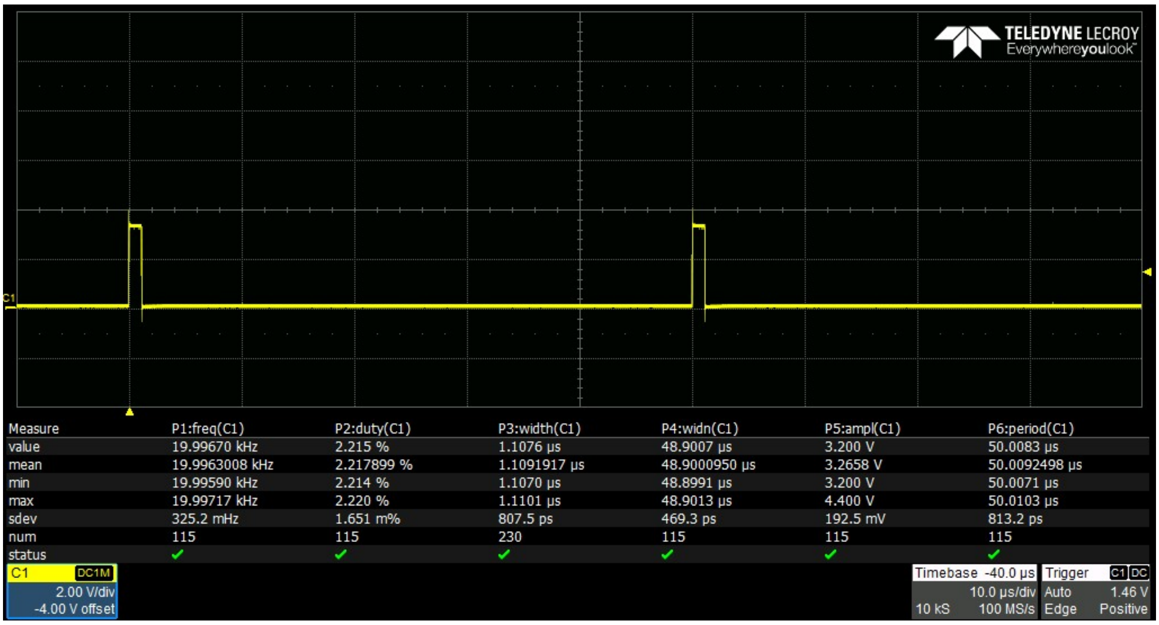
```
C1  DC1M          Timebase  20.2 µs  Trigger  C1 DC
  1.000 V/div            10.0 µs/div  Auto      1.58 V
 -2.000 V ofst    100 kS      1 GS/s  Edge    Positive
```

Figure 7.3: Park execution time using CMSIS

As shown in figure in figure 7.3, the execution time for Park transform in this configuration was about 6.2 μs.

Since this timing occupies about 12% of the machine cycle time, further analysis have been done in this module trying to reduce the computation time. Considering that the time spent to perform a long type operation is about twice the time for the integer type, while the time spent to perform a float type is 30% more than an integer type, a new version of sine and cosine functions was developed using float values with a customized lookup table, thus basing the Park transform operations on integer and float values.

The result is a reduction of the computation time for Park transform. From a second test, the measured computation time was about 1.1 μs, as shown in figure 7.4.



Measure	P1:freq(C1)	P2:duty(C1)	P3:width(C1)	P4:widn(C1)	P5:ampl(C1)	P6:period(C1)
value	19.99670 kHz	2.215 %	1.1076 µs	48.9007 µs	3.200 V	50.0083 µs
mean	19.9963008 kHz	2.217899 %	1.1091917 µs	48.9000950 µs	3.2658 V	50.0092498 µs
min	19.99590 kHz	2.214 %	1.1070 µs	48.8991 µs	3.200 V	50.0071 µs
max	19.99717 kHz	2.220 %	1.1101 µs	48.9013 µs	4.400 V	50.0103 µs
sdev	325.2 mHz	1.651 m%	807.5 ps	469.3 ps	192.5 mV	813.2 ps
num	115	115	230	115	115	115
status	✓	✓	✓	✓	✓	✓

C1 DC1M
2.00 V/div
-4.00 V offset

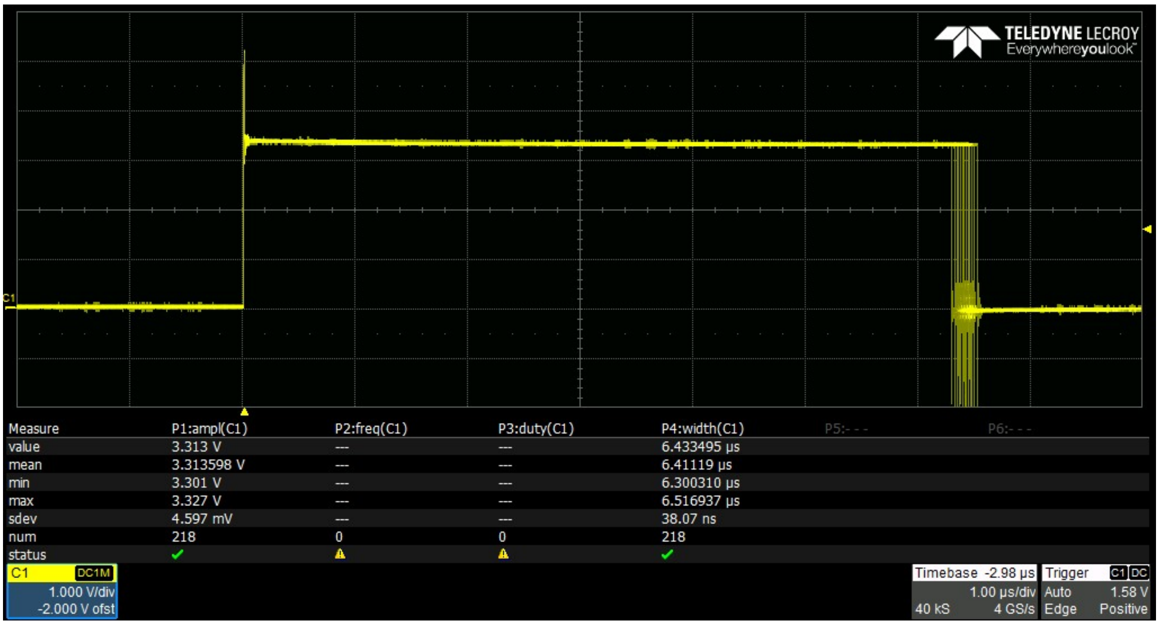
Timebase -40.0 µs
10.0 µs/div
10 kS 100 MS/s

Trigger C1 DC
Auto 1.46 V
Edge Positive

Figure 7.4: Improved Park execution time

7.2.2 Cartesian to Polar Transform Execution Time

The cartesian to polar transform module was also subject of an analysis with the objective to reduce the computation time of the overall FOC algorithm.
In fact, the first version of the module was implemented using an exact square root function and float/double data type, thus leading to an execution time of about 6.4 μs, as shown in figure 7.5.
To speed up the operations, the double values have been replaced with floats and a new square root function has been implemented. The new square root function lead to a faster computation time, at the cost of loss of precision, which considering the kind of data results acceptable. A new test of the improved version of the cartesian to polar transform module was made, which highlighted the improvements showing an execution time of about 1.6 μs, as shown in figure 7.6.



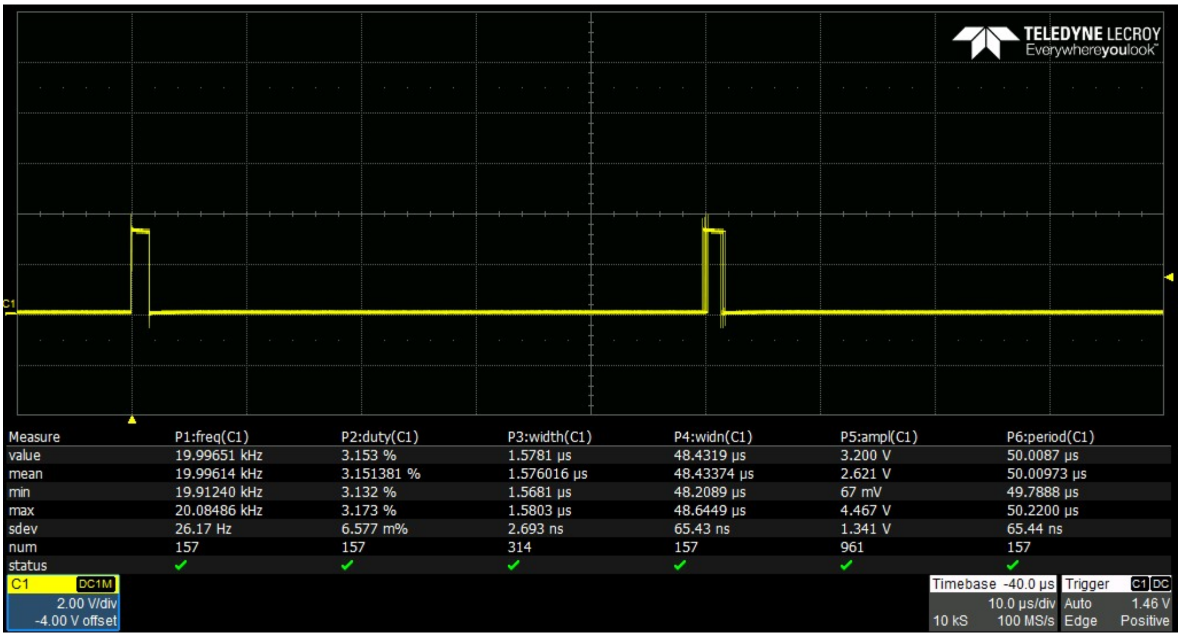
Measure	P1:ampl(C1)	P2:freq(C1)	P3:duty(C1)	P4:width(C1)	P5:- - -	P6:- - -
value	3.313 V	---	---	6.433495 µs		
mean	3.313598 V	---	---	6.41119 µs		
min	3.301 V	---	---	6.300310 µs		
max	3.327 V	---	---	6.516937 µs		
sdev	4.597 mV	---	---	38.07 ns		
num	218	0	0	218		
status	✓	⚠	⚠	✓		

C1 DC1M
1.000 V/div
-2.000 V ofst

Timebase -2.98 µs
1.00 µs/div
40 kS 4 GS/s

Trigger C1 DC
Auto 1.58 V
Edge Positive

Figure 7.5: Cart to Polar transform execution time



Measure	P1:freq(C1)	P2:duty(C1)	P3:width(C1)	P4:widn(C1)	P5:ampl(C1)	P6:period(C1)
value	19.99651 kHz	3.153 %	1.5781 µs	48.4319 µs	3.200 V	50.0087 µs
mean	19.99614 kHz	3.151381 %	1.576016 µs	48.43374 µs	2.621 V	50.00973 µs
min	19.91240 kHz	3.132 %	1.5681 µs	48.2089 µs	67 mV	49.7888 µs
max	20.08486 kHz	3.173 %	1.5803 µs	48.6449 µs	4.467 V	50.2200 µs
sdev	26.17 Hz	6.577 m%	2.693 ns	65.43 ns	1.341 V	65.44 ns
num	157	157	314	157	961	157
status	✓	✓	✓	✓	✓	✓

C1 DC1M
2.00 V/div
-4.00 V offset

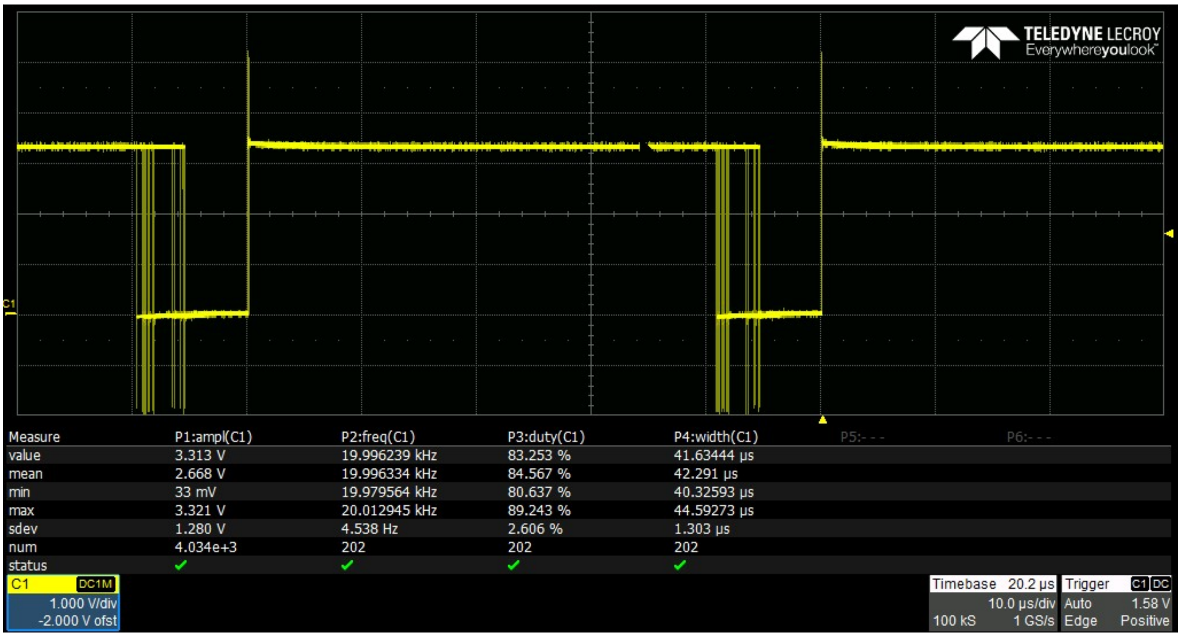
Timebase -40.0 µs
10.0 µs/div
10 kS 100 MS/s

Trigger C1 DC
Auto 1.46 V
Edge Positive

Figure 7.6: Improved Cart to Polar transform execution time

7.2.3 Machine Cycle Execution Time

Finally, in this section it is presented the total machine execution time, starting from a first version of the FOC algorithm without the improvements shown in sections 7.2.1 and 7.2.2.
This first version of the firmware was implemented using CMSIS library, which use 32-bit and 64-bit data types. The measurement of the machine cycle execution time was about 44.6 μs, as shown in figure 7.7.



Measure	P1:ampl(C1)	P2:freq(C1)	P3:duty(C1)	P4:width(C1)	P5:- - -	P6:- - -
value	3.313 V	19.996239 kHz	83.253 %	41.63444 µs		
mean	2.668 V	19.996334 kHz	84.567 %	42.291 µs		
min	33 mV	19.979564 kHz	80.637 %	40.32593 µs		
max	3.321 V	20.012945 kHz	89.243 %	44.59273 µs		
sdev	1.280 V	4.538 Hz	2.606 %	1.303 µs		
num	4.034e+3	202	202	202		
status	✓	✓	✓	✓		

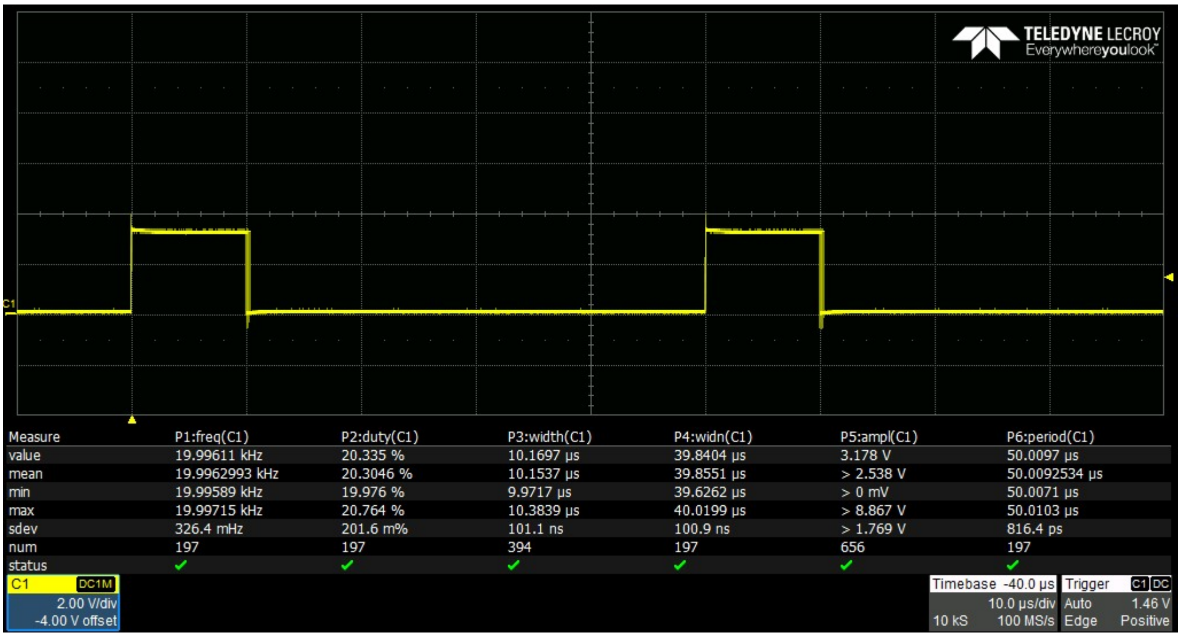
C1 DC1M
1.000 V/div
-2.000 V ofst

Timebase 20.2 µs
10.0 µs/div
100 kS 1 GS/s

Trigger C1 DC
Auto 1.58 V
Edge Positive

Figure 7.7: Machine cycle time

The CMSIS functions were replaced with the improvements presented in subsections 7.2.1 and 7.2.2, leading to an execution time of 10.22 μs, as show in figure 7.8. From these results, it has been concluded that the XMC4400 is suitable for performing 32-bit integer and float operations instead 64-bit integer operations.
With these new short execution times, the microcontroller can perform other operations while the field oriented control is running, or alternatively can control multiple motors at the same time.



Measure	P1:freq(C1)	P2:duty(C1)	P3:width(C1)	P4:widn(C1)	P5:ampl(C1)	P6:period(C1)
value	19.99611 kHz	20.335 %	10.1697 µs	39.8404 µs	3.178 V	50.0097 µs
mean	19.9962993 kHz	20.3046 %	10.1537 µs	39.8551 µs	> 2.538 V	50.0092534 µs
min	19.99589 kHz	19.976 %	9.9717 µs	39.6262 µs	> 0 mV	50.0071 µs
max	19.99715 kHz	20.764 %	10.3839 µs	40.0199 µs	> 8.867 V	50.0103 µs
sdev	326.4 mHz	201.6 m%	101.1 ns	100.9 ns	> 1.769 V	816.4 ps
num	197	197	394	197	656	197
status	✓	✓	✓	✓	✓	✓

Figure 7.8: Improved machine cycle time

8 Conclusions

The aim of the thesis was to implement a digital Field Oriented Control for Permanent Magnets Synchronous Motors using ARM-M4 architectures.
Starting from the theoretical Field Oriented Control scheme, it has been possible to implement an efficient and versatile control for any Permanent Magnet Synchronous Motor.
The theoretical scheme has been adapted to be implemented on an ARM-M4 microcontroller, taking into account practical limits, such as computing power and simplicity of implementation.
The selection of the microcontroller led to the Infineon XMC4400 which respects all the requirements necessary to the FOC to be implemented.
The implementation of many control schemes permits to control different parameters such as speed or torque.
Furthermore, the developed firmware can be adapted to multiple boards that mount XMC4400 microcontroller and to any three-phase inverter and PMSM electric motor. After analysis on the computation time of the FOC algorithm, the latter has been optimized thus leading to excellent performances in terms of computation time.
Moreover, the modularity of the developed firmware allows to add, modify or remove any module without affecting the others.
Further firmware improvements and new features can be implemented, for example, in order to improve the position and speed estimation an encoder or a hall sensor can be added.
The field-oriented control method coupled with microcontroller technology has been shown to be an effective means for controlling a three-phase permanent-magnet synchronous motor and, at the same time, it is a very efficient and low cost control.
Although field-oriented control could be implemented using analog hardware, the advent of microcontroller has eliminated many of the difficulties such as noise problems and complexities as well as cost associated with this approach. Even present day low-cost microcontrollers can easily handle the functions, primarily coordinate transformations, which are required for field-oriented control.
The control scheme is using the flux feedback signal based only on stator voltages

and currents as feedbacks but other techniques for computing the flux vector can be evaluated in an attempt to find the best and most practical solution, depending on the hardware setup and the requirements that need to be satisfied.

Although the described system was initially designed for a PWM three-phase inverter with a high modulation frequency, there is no inherent restriction limiting the application to this type of inverter. Other inverters, such as impressed current or variable DC-link voltage types could be used, possibly with some reduction in dynamic performance. In the future, with the evolution in materials, electronics and controls, the aim will be to have increasingly efficient and less expensive devices, bringing the spread of electric motors to a new level.

Bibliography

[1] *“Energy Efficiency in Motor Driven Systems 2007 Conference Reports"*, 'The Japan Electrical Manufacturers' Association, Electrical Manufacture 14 (Oct. 2007).

[2] Hiroyuki Mikami, Dr. Eng. Kazumasa Ide, Dr. Eng. Yukiaki Shimizu Masaharu Senoo Hideaki Seki: *“Historical Evolution of Motor Technology"*, Hitachi Review Vol. 60 (2011), No. 1

[3] JJ Di Steffano, AR Stubberud, IJ Williams: *“Feedback and control systems”*, McGraw-Hill, 1967.

[4] Manfre Alessandro: *“Sensorless FOC Motor Control”*, Infineon Technologies, July 2013

[5] *“AP08116 Sensorless Field Oriented Control (FOC) on XC836”*, Infineon Technologies AG, February 2011

[6] *“XMC The industrial and Multimarket MCU, Technical Introduction”*, Infineon Technologies AG, May 2016

[7] *“XMC HoT with DAVE™v4”*, Infineon Technologies, May 2016

[8] *“Field Oriented Control of Permanent Magnet Synchronous Motors”*, Microsemi, 2012

[9] *“PMSM FOC motor control software using XMC™”*, Infineon Technologies, August 2019

[10] *“XMC™ in Application - Peripherals for Motor Control”*, Infineon Technologies, 2016

[11] *“3-Phase PMSM FOC Control”*, Spansion Inc, February 2015

[12] Zhao Tao: *"Sensorless Field Oriented Control for Permanent Magnet Synchronous Motors"*, Infineon Technologies, June 2013

[13] *"XMC4400, Microcontroller Series for Industrial Applications, Reference Manual"*, Infineon Technologies, July 2016

[14] *"Hexagon Application Kit, Board User's Manual"*, Infineon Technologies, February 2013

[15] Silverio Bolognani: *"Electric Drives for Automation - Lecture notes"*, Academic year 2015/2016.

[16] Luca Corradini: *"Power Electronics - Lecture notes"*, Academic year 2015/2016.

[17] Rupprecht Gabriel, Werner Leonhard, Craig J. Nordby: *"Field-Oriented Control ofa Standard AC Motor Using Microprocessors"*, IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, VOL. IA-16, NO. 2, March/April 1980

[18] Gabriel H. Negri, Filipe G. Nazário1, José de Oliveira1, Ademir Nied, *"Back-emf Based Rotor Position Estimation For Low Cost Pmsm Drive Using Fully Connected Cascade Artificial Neural Networks"*, Eletrôn. Potên., Campo Grande, v. 23, n. 1, p. 69-77, jan./mar. 2018